A Deep Dive into Computer Vision Aided Sewer Inspections

Haurum, Joakim Bruslund

DOI (link to publication from Publisher):
10.54337/aau478559005

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

A DEEP DIVE INTO COMPUTER VISION AIDED SEWER INSPECTIONS

BY
JOAKIM BRUSLUND HAURUM

DISSERTATION SUBMITTED 2022

AALBORG UNIVERSITY
DENMARK

A Deep Dive into Computer Vision Aided Sewer Inspections

Ph.D. Dissertation

Joakim Bruslund Haurum

Dissertation submitted April 1, 2022

Dissertation submitted:	April 1, 2022
PhD supervisor:	Professor Thomas B. Moeslund Aalborg University
PhD committee:	Associate Professor Georgios A Triantafyllidis (chair) Aalborg University, Denmark Professor Serge Belongie University of Copenhagen, Denmark Professor Graham W. Taylor University of Guelph, Canada
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-919-6

Published by:
Aalborg University Press
Kroghstræde 3
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Stibo Complete, 2022

Curriculum Vitae

Joakim Bruslund Haurum

Joakim Bruslund Haurum received his B.Sc. in Medialogy and M.Sc. in Vision, Graphics, and Interactive Systems from Aalborg University, Denmark, in 2016 and 2018, respectively. During his M.Sc. studies, he worked as a student assistant in the Visual Analysis and Perception (VAP) Laboratory, as a student teacher for Section of Media Technology, Aalborg University and as a software developer intern at Unity Studios ApS. After working 6 months as a research assistant he commenced his Ph.D. studies in the VAP Lab in January 2019 at the Department of Architecture, Design and Media Technology, Aalborg University. During his Ph.D., he served on the Ph.D. Board of the Technical Doctoral School of IT and Design at Aalborg University for nearly three years, including two years as vice-chairman of the board.

As part of his Ph.D. studies he has collaborated with the Human Pose Recovery and Behavior Analysis (HuPBA) research group at the Universitat de Barcelona & the Computer Vision Center at the Autonomous University of Barcelona. The collaboration was conducted over a nine month period as primarily virtual collaboration, with a one week visit in October 2021 at the HuPBA research group in Barcelona, Spain.

His main interest are computer vision, machine learning, and computer graphics, especially in the area of automating complex manual tasks such as sewer inspections. He has been involved in the supervision of undergraduate and graduate projects within image processing, robotics, computer graphics, and computer vision.

Curriculum Vitae

Abstract

The sewerage infrastructure is a critical infrastructure of modern society, which requires regular inspections. However, due to the large extent of the infrastructure it is infeasible to inspect all parts regularly through manual inspections. This Ph.D. thesis addresses the topic of computer vision aided automation of sewer inspections through two input modalities: images and point clouds.

Within image-based automation of sewer inspection, we investigated the fundamental historic hindrances of the research field, and how the field can be advanced. Through a survey covering nearly three decades, we found that the research field was lagging behind the general computer vision field by several years and pinpointed three major hindrances: A lack of public data, open-source code, and a common evaluation protocol. Using data from three Danish water utility companies, we released the world's first publicly available sewer multi-label defect classification dataset: Sewer-ML. Using Sewer-ML we benchmarked 12 state-of-the-art algorithms implemented in an open-source codebase, evaluated using two domain-influenced evaluation metrics. Through this analysis, we documented the need for further research in the field.

We advanced the image-based automation of sewer inspections field by first considering the equally important tasks of water level, pipe material, and pipe shape classification. An initial investigation using a subset of Sewer-ML and common computer vision models found that the water level in sewer pipes is better classified when using water level labels based on visual appearances compared to exact quantities. Building upon this result, we demonstrated the effectiveness of a multi-task classification approach for classifying all four tasks at once and presented a method to improve performance by incorporating known relationships between classes across tasks. We also extended the recent Hybrid Vision Transformer with multi-scale features and a clustering-based tokenizer in order to capture the spatial semantics of sewer defects, achieving significant improvements within sewer defect classification.

Within the point cloud-based automation of sewer inspections field we presented a synthetic sewer point cloud generator to circumvent the lack of real life data. Using the synthetic data generator and data recorded from a laboratory setup, we released the world's first point cloud-based dataset for sewer defect classification and compared performance of the PoitnNet and DGCNN models. Through this analysis, we verified the usefulness of synthetic point clouds for training sewer defect classification models.

Resumé

Vand- og spildevandsinfrastrukturen er en kritisk infrastruktur i moderne samfund som kræver regelmæssige inspektioner. Det er dog umuligt at inspicere hele infrastrukturen regelmæssigt grundet infrastrukturens store omfang. Denne Ph.D.-afhandling omhandler brugen af computer vision til at automatisere kloakinspektioner ved brug af to input modaliteter: billeder og punktskyer.

Inden for billede-baseret automatisering af kloakinspektioner undersøgte vi de fundamentale historiske forhindringer i forskningsområdet samt hvordan forskningsområdet kan avanceres. Ved at undersøge de seneste tre årtier fandt vi ud af at forskningsfeltet oplever en flerårig forsinkelse i forhold til det generelle computer vision-felt og herfra udpeger vi tre store hindringer: En mangel på offentligt data, open-source kode, og en fælles evalueringsprotokol. Ved brug af data fra tre danske vandforsyninger udgav vi verdens første offentlige kloakfejl klassificerings datasæt: Sewer-ML. Vi sammenlignede 12 state-of-the-art algoritmer implementeret i en open-source kodebase, som vi evaluerede ved brug af to domæne-inspirerede evalueringsmetrikker. Via denne analyse dokumenterede vi behovet for videre forskning i feltet.

Vi fremmede forskningsfeltet inden for billede-baseret automatisering af kloakinspektioner ved først at undersøge klassificering af vandniveau, rørmateriale, og rørform. I en indledende undersøgelse der brugte dele af Sewer-ML datasættet og almindelige computer vision modeller fandt vi at vandniveauet i et kloakrør er bedre klassificeret når annoteringer er baseret på visuel udseende end den procentvise vandstand. Derefter demonstrerede vi effektiviteten af en multi-task klassificeringstilgang på de fire opgaver samtidigt og præsenterede en metode til at forbedre præstationen ved at gøre brug af kendte forhold mellem klasserne i de fire opgaver. Vi forbedrede klassificering af kloakfejl ved at tilføje multi-scale features og en clustering-baseret tokenizer til Hybrid Vision Transformer modellen for at udnytte de rumlige relationer mellem kloakfejl.

Inden for forskningsfeltet omhandlende punktsky-baseret automatisering af kloakinspektioner præsenterede vi en syntetisk kloak punktsky-generator til at omgå manglen på data fra det virkelige liv. Ved brug af den syntetiske data generator og data optaget i et laboratorieopsætning udgav vi verdens første punktsky-baserede datasæt til klassificering af kloakfejl og sammenlignede præstationen af PointNet og DGCNN modellerne. Gennem denne analyse verificerede vi brugbarheden af syntetiske punktskyer til at træne modeller der kan klassificere kloakfejl.

Contents

IV Point Cloud-based Automation of Sewer Inspections 281

Thesis Details

Thesis Title: A Deep Dive into Computer Vision Aided Sewer Inspections
Ph.D. Student: Joakim Bruslund Haurum
Supervisor: Professor Thomas B. Moeslund, Aalborg University

The main body of this thesis consists of the following papers.

[A] Joakim Bruslund Haurum, Thomas B. Moeslund, “A Survey on Image-Based Automation of CCTV and SSET Sewer Inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[B] Joakim Bruslund Haurum, Thomas B. Moeslund, “Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13451–13462, 2021.

[C] Joakim Bruslund Haurum, Chris H. Bahnsen, Malte Pedersen, Thomas B. Moeslund, “Water Level Estimation in Sewer Pipes Using Deep Convolutional Neural Networks,” *Water*, vol. 12, no. 12, p. 3412, 2020.

[D] Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, Thomas B. Moeslund, “Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder,” *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1441–1452, 2022.

[E] Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, Thomas B. Moeslund, “MSHViT: Multi-Scale Hybrid Vision Transformer and Sinkhorn Tokenizer for Sewer Defect Classification,” Submitted to *Machine Vision and Applications*, 2022.

[F] Kasper Schøn Henriksen, Mathias S. Lynge, Mikkel D. B. Jeppesen, Moaaz M. J. Allahham, Ivan A. Nikolov, Joakim Bruslund Haurum, Thomas B. Moeslund, “Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation,” *Proceedings of the 7th International Conference on Augmented Reality, Virtual Reality and Computer Graphics (SalentoAVR)*, pp. 364—373, 2020

[G] Joakim Bruslund Haurum, Moaaz M. J. Allahham, Mathias S. Lynge, Kasper Schøn Henriksen, Ivan A. Nikolov, Thomas B. Moeslund, "Sewer Defect Classification using Synthetic Point Clouds," *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 891—900, 2021

In addition to the main papers, the author has also been involved in the following publications. These publications are not included in the thesis as they are found to be out of scope.

- Joakim Bruslund Haurum, Chris H. Bahnsen, Thomas B. Moeslund, "Is it Raining Outside? Detection of Rainfall using General-Purpose Surveillance Cameras," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

- Malte Pedersen, Joakim Bruslund Haurum, Rikke Gade, Thomas B. Moeslund, Niels Madsen, "Detection of Marine Animals in a New Underwater Dataset with Varying Visibility," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019.

- Joakim Bruslund Haurum, Anastasija Karpova, Malte Pedersen, Stefan Hein Bengtson, Thomas B. Moeslund, "Re-Identification of Zebrafish using Metric Learning," *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 1–11, 2020.

- Malte Pedersen, Joakim Bruslund Haurum, Stefan Hein Bengtson, Thomas B. Moeslund, "3D-ZeF: A 3D Zebrafish Tracking Benchmark Data," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2423–2433, 2020.

- Malte Pedersen, Joakim Bruslund Haurum, Thomas B. Moeslund, Marianne Nyegaard, "Re-Identification of Giant Sunfish using Keypoint Matching," *Proceedings of the Northern Lights Deep Learning Workshop*, 2022.

This thesis has been submitted for assessment in partial fulfillment of the PhD degree. The thesis is based on the submitted or published scientific papers which are listed above. Parts of the papers are used directly or indirectly in the extended summary of the thesis. As part of the assessment, co-author statements have been made available to the assessment committee and are also available at the Faculty.

Preface

The thesis is structured as a collection of papers in partial fulfillment of a Ph.D. study at the Section of Media Technology, Aalborg University, Denmark. The scientific work covers three research topics: Foundations for Image-based Automation of Sewer Inspections, Advancing Image-based Automation of Sewer Inspections, and Point Cloud-based Automation of Sewer Inspections. The thesis is organized into four parts. Part one contains an introduction to how the manual inspection process is conducted, as well as an overview of the state-of-the-art for each of the three main research topics and contributions made to each of them. This is followed by a part for each of the three research topics containing the selected papers published during the Ph.D.

This project has been carried out from 2019-2022, mainly in the Visual Analysis and Perception (VAP) Laboratory at Aalborg University, with a predominantly virtual research stay at the Human Pose Recovery and Behavior Analysis (HuPBA) research group at the Universitat de Barcelona and the Computer Vision Center, Spain.

I would like to thank my supervisor Prof. Thomas B. Moeslund for excellent supervision throughout my Ph.D.-study and providing me with guidance and freedom to pursue my research interests and grow as a researcher, while not steering too far off course. I want to thank Chris Bahnsen for mentoring me when I first started in VAP lab, Malte Pedersen for our enjoyable research projects together, as well as all of my current and former colleagues in the VAP lab for creating a great work environment. Similarly, I want to thank Prof. Sergio Escalera and Dr. Meysam Madadi for hosting me in the HuPBA lab, for providing thoughtful guidance, and showing great interest in my research ideas. I also want to thank the Innovation Fund Denmark for funding the ASIR project, as well as our ASIR partners from the Danish water sector for readily sharing their domain expertise as well as provide the crucial data needed.

Finally, I would never have been able to complete this journey if it had not been for the support from all of my amazing friends and family, who have had to put up with me being preoccupied with the Ph.D. but also brightened my day when most needed. I am especially thankful for my wonderful girlfriend Anne, whom has given me nothing but invaluable and unwavering love and support at all times, making even the hardest of days manageable.

Joakim Bruslund Haurum
Aalborg University, April 1, 2022

Dedicated to my mother, Tina, who was taken from us far to early.
I hjertet gemt, men aldrig glemt.

Part I

Overview of the Work

Chapter 1

Introduction

A long standing goal of computer science has been to build a system which can act and behave similar to humans at any task, an Artificial General Intelligence (AGI). While the AGI is currently purely hypothetical and may never be realizable, the approximation of human traits has rooted itself in several research disciplines. One such trait, the human perception, has been a major driver of the Computer Vision research field, where the goal is to develop machines capable of extracting high-level information from optical sensors. Through the use of statistical and machine learning-based methods and large amount of data from the increasingly digital world, the computer vision field has grown exponentially within the last decade, and been applied in numerous real life applications such as in autonomous vehicles and for fingerprint and face verification on smartphone. However, the computer vision field has also had a major impact in the industrial sector by enabling automatic quality inspection processes in factories, as well as automating the inspection of critical infrastructure in society, such as the sewerage and transportation infrastructures, in order to assist professional inspectors.

This thesis is part of a larger research project called *Automated Sewer Inspection Robot (ASIR)*, focused on the automation of the entire sewer inspection process using autonomous robots “living” in the sewers. The project is funded by the Innovation Fund Denmark (grant number: 8055-00015A). Currently, sewer inspections are performed by certified inspectors maneuvering a remote-controlled robot with a steerable camera through the sewer pipes. The sewerage infrastructure has to be regularly inspected for faulty pipes, which can have major environmental and health-wise risks if sewage was to *e.g.* exfiltrate into the groundwater or overflow into the streets. However, due to the large extent of the sewerage infrastructure, it is impossible to regularly inspect all sewer pipes. This leads to pipes being replaced 15-25 years before initially projected in order to reduce the risk of breakdowns in the critical sewerage infrastructure, resulting in an increased financial cost [1]. It is estimated that if the lifetime of the sewer pipes in Denmark can be extended by just 10%, the water utility companies in Denmark can collectively expect a cost saving of more than 100 million Danish kroner per year as

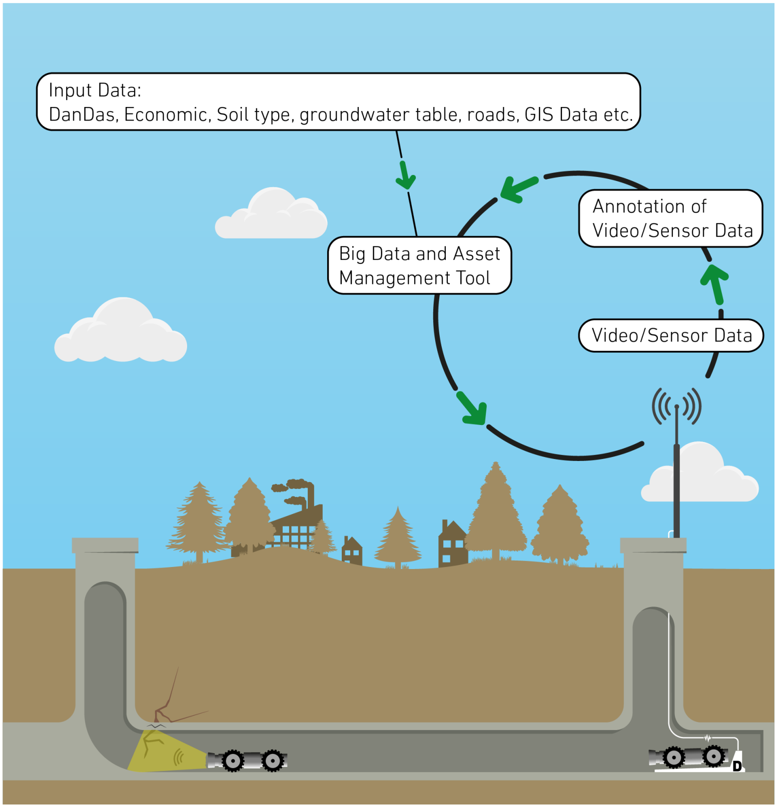

Input Data:
DanDas, Economic, Soil type, groundwater table, roads, GIS Data etc.
Big Data and Asset Management Tool
Annotation of Video/Sensor Data
Video/Sensor Data
D

Fig. 1.1: Illustration of the ASIR project. An overview of the different areas the Automated Sewer Inspection Robot research project touches upon and how they all relate. This thesis is focused on the *Annotation of Video/Sensor Data* part of the illustrated pipeline. *Copyright by Aalborg University and reproduced with permission.*

a results of the improved asset management [1]. This is the aim of the ASIR project, to increase the inspection rate of the sewer pipes using autonomous sewer inspection robots and in turn use the more densely sampled data to improve the asset management of the sewerage infrastructure, as illustrated in Figure 1.1.

By deploying autonomous robots in the sewers continuously, it is possible to cover a larger part of the infrastructure simultaneously, while also obtaining a denser sampling of the sewerage infrastructure deterioration status. This enables a more fine-grained asset management of the infrastructure and thereby better allocation of the utility companies' budgets. However, the development of such autonomous robots are non-trivial as custom hardware platforms, navigation systems, and inspection algorithms have to be developed. Several attempts to construct such robots have been made [2–8], with only few designs making it to the commercial market [9]. Even then, the current commercial products are not fully autonomous robots as they still depend on aspects such as manual deployment and do not automatically determine the sewer state.

In the context of the ASIR project, the work in this PhD focuses on the development of algorithms that can automate the sewer inspection process using computer vision. Specifically, the work focuses on the study of using the currently available image data and point cloud data from a time-of-flight sensor selected as a part of the ASIR project for algorithm development. Through these studies we shed light on the current status of the automatic sewer inspection field uncovering systematic problems, which we analyze in depth and propose concrete solutions to improve and further advance he field.

1 Image-based Automation of Sewer Inspections

The main sensor used for sewer inspections is a monocular camera observing the visible spectrum, an approach called Closed-Circuit Television (CCTV). This approach lends itself well to inspections, as the data is easy to interpret on-site by the inspector and can be stored for later verification off-site. Image-based automation approaches have, therefore, been the primary focus of the research field for nearly 30 years due to the ease of access to data compared to other sensors [10]. However, due to commercial interests data and code have rarely been publicly available, limiting the advances within the field and enforcing a high-barrier of entrance.

This has limited the ability to develop generalized solutions, and instead directed the field to focus on developing solutions for specific utility companies, sometimes further limited to small sections in cities. While these case studies may be “solved” they are not guaranteed to generalize across city, country, or continental borders as the sewerage infrastructures differs due to differences in construction of the sewerage infrastructure and low number of data samples. Nonetheless, there are several companies developing more generalized solutions in collaboration with utility companies [11–14]. Furthermore, current solutions primarily focus on the classification and detection of defects, neglecting other parts of the sewer inspection process such as identifying the pipe properties (material, shape, and diameter) needed to accurately determine the level of deterioration in the pipe.

This thesis presents work on quantifying the effect of recent advances, democratizing the access to sewer inspection data and novel approaches to enabling image-based automation of sewer inspections are presented.

2 Point Cloud-based Automation of Sewer Inspections

Alternative sensing options have long been of great interest for automation of sewer inspections [15, 16]. The primary focus of this research branch has been on the gathering and processing of 3D information, be it through laser scanners, stereo cameras, or depth cameras, while the use of ultrasound, sonar, and thermal sensors has been experimented with as well. These alternative modalities can provide valuable

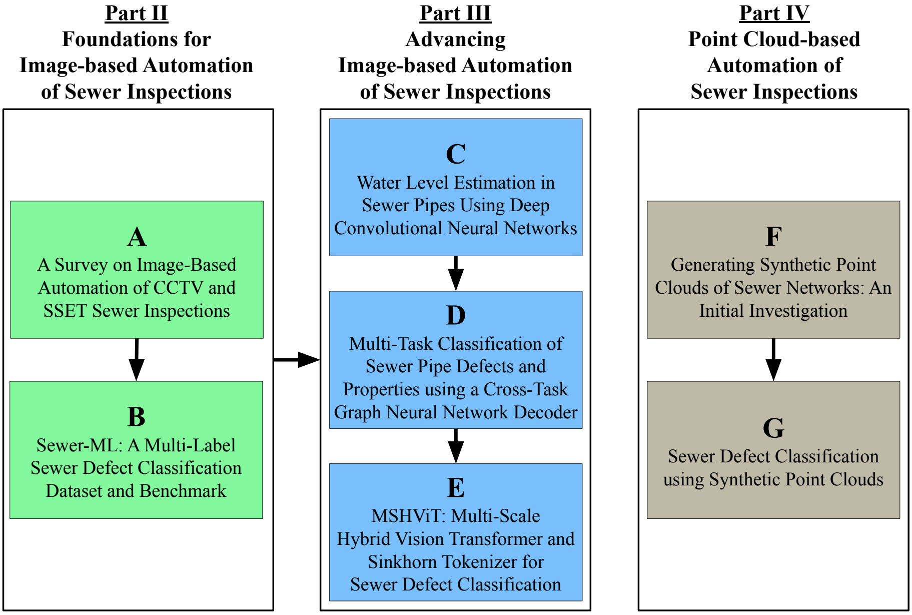

Part II
Foundations for
Image-based Automation
of Sewer Inspections
A
A Survey on Image-Based Automation of CCTV and SSET Sewer Inspections
B
Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark
Part III
Advancing
Image-based Automation
of Sewer Inspections
C
Water Level Estimation in Sewer Pipes Using Deep Convolutional Neural Networks
D
Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder
E
MSHViT: Multi-Scale Hybrid Vision Transformer and Sinkhorn Tokenizer for Sewer Defect Classification
Part IV
Point Cloud-based
Automation of
Sewer Inspections
F
Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation
G
Sewer Defect Classification using Synthetic Point Clouds

Fig. 1.2: Overview of scientific work. An overview of how the scientific work within this thesis relates to each other. Each paper is represented by a box colored according to the corresponding thesis part, with the capital letter referring to the corresponding appendix.

information that is unclear or unavailable from standard CCTV sensing, allowing easier identification of *e.g.* structural defects through depth information, water infiltrating into the pipe through the use of thermal information, and 3D reconstructions of the pipes. However, the use of these alternative sensing approaches has so far been largely limited during manual inspections due to the output being hard to interpret. As the sensors are not used in the manual inspections there is currently significantly less data from alternative sensors compared to the CCTV sensor, thereby hindering the feasibility of developing generalized automation solutions. Therefore, solutions have been primarily restricted to case studies based on data gathered using custom designed hardware, resulting in a data collection procedure that is even more restricted than the one used in the research on image-based automation of sewer inspections.

This thesis presents work on synthetically generating sewer point clouds to ease the data collection process, as well as a proof-of-concept validation of using a combination of synthetic and real data for point cloud-based automation of sewer inspections.

3 Thesis Structure

This thesis is divided into four main parts. In the following chapters, we will introduce different ways of inspecting sewer pipes. First, the current manual sewer inspection process is described and discussed in detail in Chapter 2. This chapter is included to provide the reader with the context regarding how sewer inspections are performed, and does not include any new scientific contributions.

With the context of sewer inspections established, Chapter 3-5 provide introductions to image-based and point cloud-based automation of sewer inspection, respectively, where the state-of-the-art and our conducted research is presented and discussed. Each chapter is summarized by a set of sub-conclusions and contributions, and contains separate bibliographies. Two chapters are dedicated to the image-based automation of sewer inspection field: Chapter 3 is focused on determining and overcoming fundamental historic hindrances in the field, whereas Chapter 4 is centered on the study of advancing the state of the image-based automation field. Chapter 5 is focused on the promising research direction of using depth based data, specifically point clouds, to automating the sewer inspection process and the related challenges. Lastly, Chapter 6 summarizes the work done and discusses future directions in the automated sewer inspections research field.

This is followed by Part II-IV, which are appendices containing the published papers related to image-based and point cloud-based automation of sewer pipe inspections. The relation between the papers are shown in Figure 1.2. All papers were written during the PhD study.

References

[1] Envidan, “Asir udviklingsprojekt,” accessed: 20/3-2022. [Online]. Available: https://www.envidan.dk/cases/asir-udviklingsprojekt

[2] D. Alejo, G. Mier, C. Marques, F. Caballero, L. Merino, and P. Alvito, *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*. Cham: Springer International Publishing, 2020, pp. 275–296.

[3] D. Alejo, F. Chataigner, D. Serrano, L. Merino, and F. Caballero, “Into the dirt: Datasets of sewer networks with aerial and ground platforms,” *Journal of Field Robotics*, vol. 38, no. 1, pp. 105–120, 2021.

[4] A. A. F. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, “Concept and design of a fully autonomous sewer pipe inspection mobile robot "kantaro",” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 136–143.

[5] C. H. Bahnsen, A. S. Johansen, M. P. Philipsen, J. W. Henriksen, K. Nasrollahi, and T. B. Moeslund, “3d sensors for sewer inspection: A quantitative review and analysis,” *Sensors*, vol. 21, no. 7, 2021.

[6] M. Kolesnik and H. Streich, “Visual orientation and motion control of makro-adaptation to the sewer environment,” in *In Proceedings of the Seventh International Conference on the Simulation of Adaptive Behavior*, vol. 4, 2002, pp. 62–69.

[7] F. Kirchner and J. Hertzberg, “A prototype study of an autonomous robot platform for sewerage system maintenance,” *Autonomous Robots*, vol. 4, pp. 319–331, 1997.

[8] F. Chataigner, P. Cavestany, M. Soler, C. Rizzo, J.-P. Gonzalez, C. Bosch, J. Gibert, A. Torrente, R. Gomez, and D. Serrano, *ARSI: An Aerial Robot for Sewer Inspection*. Cham: Springer International Publishing, 2020, pp. 249–274.

[9] J. M. Mirats Tur and W. Garthwaite, “Robotic devices for water main in-pipe inspection: A survey,” *Journal of Field Robotics*, vol. 27, no. 4, pp. 491–508, 2010.

[10] J. B. Haurum and T. B. Moeslund, “A survey on image-based automation of cctv and sset sewer inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[11] InLoc Robotics, “Sewdef: Automatic detection of defects in sewer networks,” accessed: 13/3-2022. [Online]. Available: https://inlocrobotics.com/en/sewdef-en/

[12] Subterra, “Terralytics,” accessed: 13/3-2022. [Online]. Available: https://www.subterra.ai/terralytics

[13] SewerAI, “Autocode,” accessed: 13/3-2022. [Online]. Available: https://www.sewerai.com/autocode

[14] Hades Technologies, “Hades ai,” accessed: 13/3-2022. [Online]. Available: https://www.hades.ai/

[15] O. Duran, K. Althoefer, and L. D. Seneviratne, “State of the art in sensor technologies for sewer inspection,” *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, April 2002.

[16] Z. Liu and Y. Kleiner, “State of the art review of inspection technologies for condition assessment of water pipes,” *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

Chapter 2

Manual Sewer Inspections

This chapter provides an introduction to how manual sewer inspections are currently performed using CCTV sensors and remote-controlled vehicles, the most commonly used inspection standards for sewer inspections, and a deeper look at the Danish inspection standard, the “Fotomanual”.

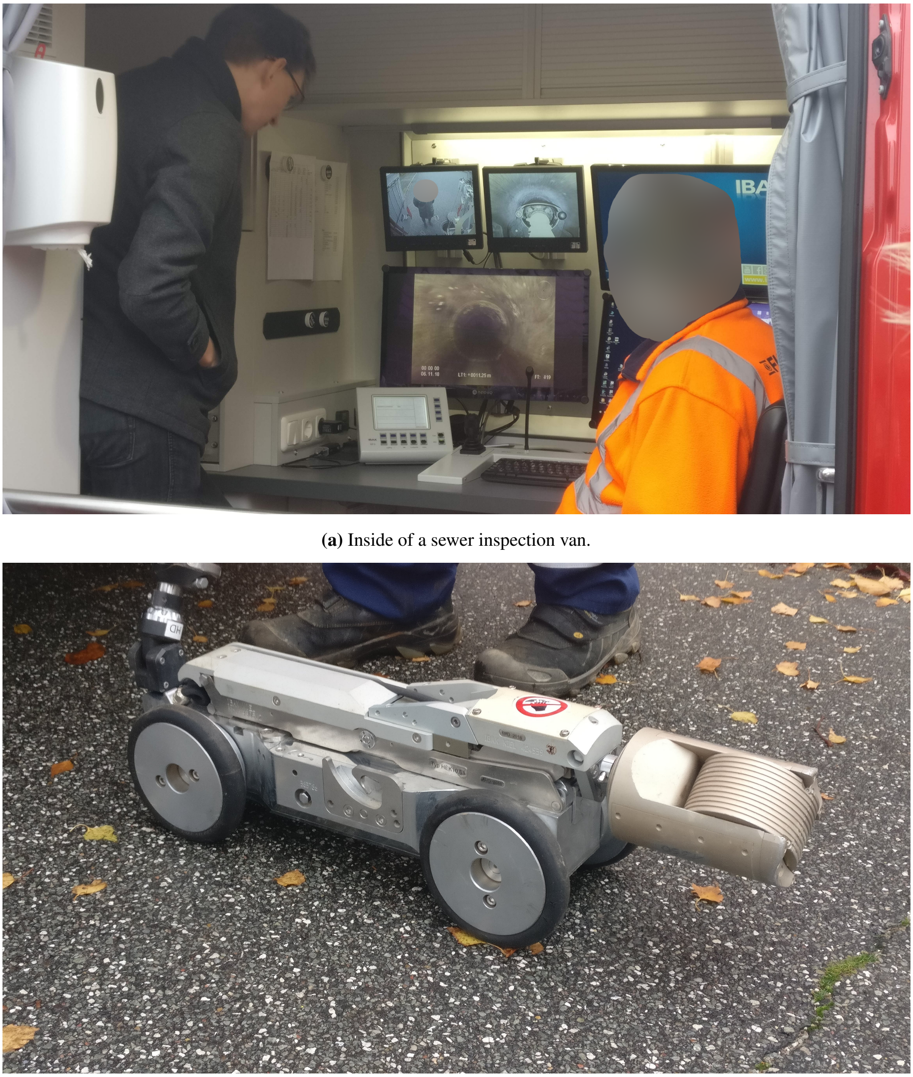
1 CCTV Sewer Inspections

The sewerage infrastructure is predominantly inspected using a method called a TV or CCTV inspection. This approach uses a remote-controlled vehicle, called a *tracktor*, controlled by the inspector above ground in the inspection van, see Figure 2.1a. The tracktor is inserted into the main[1] pipes through sewer wells, from where both main and lateral[2] sewer pipes are inspected. The inspection is performed by the inspector maneuvering the tracktor through the pipes, while simultaneously observing and logging any observed defects. This processes can be complicated by factors such as haze or mist in the sewer limiting the visibility in the sewer, poor lightning conditions in the sewer due to reflections or hardware limitations, or obstructions in the pipe requiring the inspector to extract the tracktor and processed from a well longer up the pipe. The video feed is saved and stored with all information (pipe properties, distance, time, pipe number, *etc.*) overlaid onto the screen. Lastly, the inspection of a sewer pipe is summarized into a single deterioration score, used by utility companies for asset management, based on the amount of defects as well as their type, severity, and frequencies.

The tracktor is highly modifiable in order to accommodate different sized and shaped sewer pipes, see Figure 2.1b. The tracktor is tethered to the inspection van through a power cable, which doubles as a distance meter by keeping track of the amount of cable pulled along. A CCTV sensor on the front of the tracktor capable of rotating in a hemisphere allows the inspector to adjust the viewing angle and

[1] The pipes which collect wastewater from buildings and transports it to the wastewater treatment plant.

[2] The pipes which connect buildings to a main pipe.



(a) Inside of a sewer inspection van.

(b) Example of a sewer inspection tracktor.

Fig. 2.1: Sewer inspection equipment. Example of sewer inspection van and tracktor used by FKSSlamson in Denmark. *Images taken during the ASIR kick-off meeting*.

closer inspect regions of interest. The camera is surrounded by light sources ensuring the pipe is illuminated. As the tracktors can rarely move into lateral pipes, the camera can be pushed forward into the lateral pipes using a “push rod“. In some cases there may also be extra sensors such as a camera oriented backwards, which can help classify defects that may otherwise be obscured from the forward-pointing camera.

In Denmark, the inspectors are professionally trained to follow the standard set forth by the Danish Water and Wastewater Association (DANVA) and the Danish TV-Inspection Assessment System (DTVK). A central theme of the Danish inspection approach is that everything observed in the pipe has to be documented, to ensure a complete documentation of the infrastructure. Furthermore, the quality of the inspections are regularly evaluated (1.5-3 months) through random sampling of the inspector's completed inspections [1]. Without a license from DTVK it is practically impossible to perform inspections for water utility companies or construction work, due to the large amount of trust put in the quality control work by DTVK.

2 Sewer Inspection Standards

A key part of the sewer inspection process, is the inspection standard which defines what and how the inspector should conduct the sewer inspection. While the underlying defects and other noteworthy observations are fundamentally the same, there are different approaches to how the defects are categorized and described.

The primarily used inspection standards are the Manual of Sewer Condition Classification (MSCC) from the British Water Research Center (WRc), the Pipeline Assessment Certification Program (PACP) from the American National Association of Sewer Service Companies (NASSCO), and the European standard EN 13508-2 [2]. PACP is an adaption of the MSCC used across North and South America, whereas the European standard is an amalgamation of the different standards in the European countries.

Specifically, the European standard was initially based on the Dutch standard, but was expanded to cover all standards used in Europe [3]. This led to a more complex inspection standard, due to an increase in defect classes, including at times two different description of the same class, and several ways of describing how defects can be measured. Furthermore, each country is allowed to create a national annex detailing which defect classes and descriptions are used, while retaining compatibility with the European standard, further complicating the inspection process [3].

Dirksen *et al.* [3] conducted an analysis of six case studies on the quality of the inspection process from four countries (The Netherlands, Germany, France, and Austria) covering three scenarios: results of a sewer inspection exam, interpretation of the same inspection reports, and day to day sewer inspections. Based on these case studies it was determined that when using the European standard defects were not detected 25-50% of the time, while defects were incorrectly detected only around 4% of the times. These findings were echoed by Van der Steen *et al.* [4], who compared using the old Dutch national standard with the Dutch annex of the European standard. Van der Steen *et al.* found that using the more complex European standard led to a significant increase in missed defects for several defect classes. This was attributed to overly specific descriptions in the European standard causing the inspectors to instead choose other more broad and vague defect classes. In order to improve the sewer inspection process, both Dirksen *et al.* and Van der Steen *et al.* recommend that future standards reduce the complexity of the defect descriptions as well as include photographs of the defect classes for visual reference.

Both of these recommendations are adhered to in the Danish national annex, the “Fotomanual”, while also adhering to the guideline of keeping the defect descriptions as objective as possible. This is realized by not requiring the inspectors to estimate *e.g.* length or width of cracks or breaks, in contrast to the European standard.

In the ASIR project the focus has been primarily on the Danish sewerage infrastructure, and all available sewer inspection data has been conducted using the Danish standard. Therefore, the Fotomanual will be described in more detail.

3 The Fotomanual

Sewer inspections made in Denmark are all conducted according to the Danish standards described in the Fotomanual. In this document the entire tv-inspection process is described specifying aspects such as the defect descriptions and characterizations (with both text and images), how to log all observations, and how to convert the inspection logs to the European standard.

When conducting a sewer inspection in Denmark, the Fotomanual dictates specifically what, how, and when observation should be logged. Before an inspection starts all the metadata of the inspection is logged. This metadata includes information such as:

- When was the inspection conducted.
- The unique report number and pipe ID number.
- The weather condition during the inspection.
- The equipment used for the inspection.
- Whether it is a lateral or main pipe.
- How the pipe is integrated into the sewerage infrastructure (is it *e.g.* a combined or storm flood pipe).
- Whether the inspection was performed up- or downstream.
- Whether the pipe was high pressure flushed before the inspection.

Using this metadata as a prior it is possible to understand the conditions of the inspections and if there are any biases present. For example, if the pipe was conducted downstream it may be hard to assess the condition of a pipe joint, as the joint is constructed such that water cannot escape when flowing downstream. Similarly, if the pipe has been flushed beforehand it is difficult to assess whether roots are a problem, since the high pressure flushing process is commonly combined with a root cutting process.

During the inspection of the actual pipe, the inspector has to log the condition of the pipe. Specifically, there are four aspects of the pipe which the inspector has to consider: Defects, and the pipe properties (pipe material, shape, and diameter).

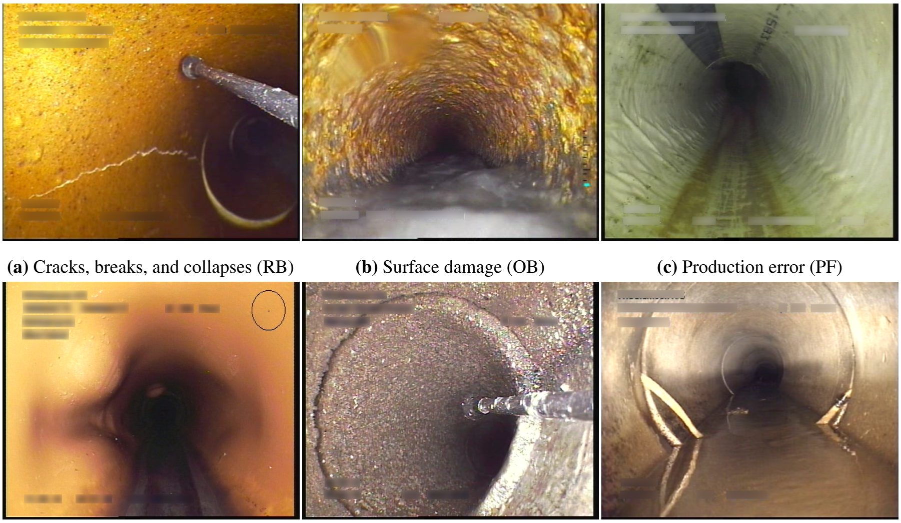
Table 2.1: Overview of the hierarchical structure of the 6th edition Fotomanual defect categories [5]. The defects are split into four super categories, each containing a set of defect categories with an associated two letter code. Each defect category is accompanied by a set of severity levels and a set of type indicators. #Severity Levels and #Types denote the number of severity levels and type indicators per defect category, respectively. *Adapted from the Fotomanual [5]*

Code	Defect Category	#Severity Levels	#Types
	Water Level		
VA	Water Level (in percentages)	11	-
	Physical Condition of the Pipe		
RB	Cracks, breaks, and collapses	4	5
OB	Surface damage	4	-
PF	Production error	4	10
DE	Deformation	4	4
FS	Displaced joint	4	3
IS	Intruding sealing material	4	2
	Operational Condition		
RØ/RO	Roots	4	3
IN	Infiltration	4	2
AF	Settled deposits	4	5
BE	Attached deposits	4	6
FO	Obstacle	4	7
	Special Constructions		
GR	Branch pipe	2	4
PH	Chiseled connection	4	4
PB	Drilled connection	4	4
OS	Lateral reinstatement cuts	4	4
OP	Connection with transition profile	4	5
OK	Connection with construction changes	5	4

3.1 Defects

An essential part of the sewer inception process is logging any defect which are encountered throughout the inspection. In the Fotomanual the defects are structured in a hierarchical manner with four super categories, 18 defect categories, as well as defect category specific severity levels and type indicators. An overview of this structure based on the 6th edition of the Fotomanual [5] is shown in Table 2.1.

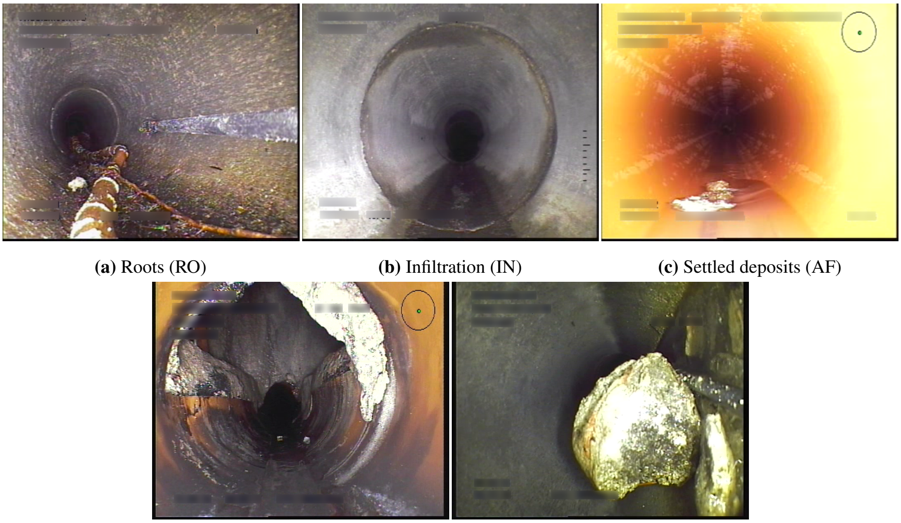
The four super categories cluster the defect categories based on how the defect originates. The *Physical Condition of the Pipe* super category contains defect categories where the defect affects the pipe itself, such as a crack in the pipe wall (RB) or the



(a) Cracks, breaks, and collapses (RB) **(b)** Surface damage (OB) **(c)** Production error (PF)

(d) Deformation (DE) **(e)** Displaced joint (FS) **(f)** Intruding sealing material (IS)

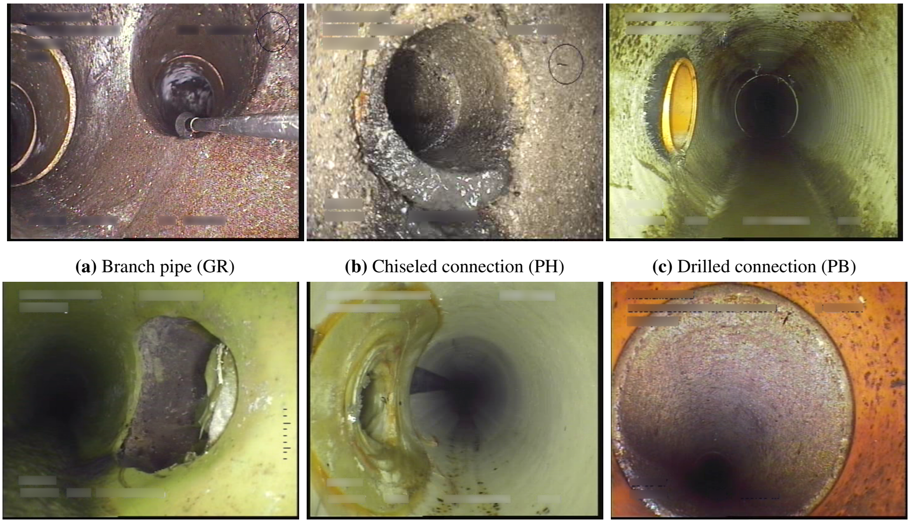
Fig. 2.2: Physical Condition of the Pipe. Examples of the defect categories in the Physical Condition of the Pipe supercategory. *Images from the Sewer-ML dataset [6].*



(a) Roots (RO) **(b)** Infiltration (IN) **(c)** Settled deposits (AF)

(d) Attached deposits (BE) **(e)** Obstacle (FO)

Fig. 2.3: Operational Condition. Examples of the defect categories in the Operational Condition supercategory. *Images from the Sewer-ML dataset [6].*

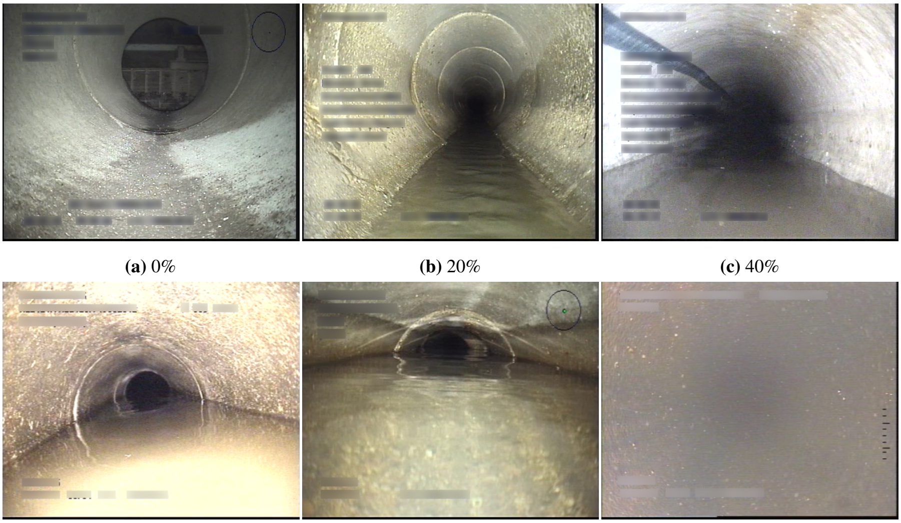


(a) Branch pipe (GR) **(b)** Chiseled connection (PH) **(c)** Drilled connection (PB)

(d) Lateral reinstatement cuts (OS) **(e)** Connection with transition profile (OP) **(f)** Connection with construction changes (OK)

Fig. 2.4: Special Construction. Examples of the defect categories in the Special Construction supercategory. *Images from the Sewer-ML dataset [6].*

sealing material from a joint hanging into the pipe (IS). In contrast the defect categories in the *Operational Condition* super category describe defects where objects or material affects the pipe, such as intruding roots (RO) or any foreign object in the pipe (FO). The *Special Constructions* super category covers all defect categories that are related to the construction of the pipes, such as when there is a change in material/shape/diameter (OK). The Special Constructions super category is special in that the defect categories are not necessarily defects, but may also be correctly made connections and transitions, which are still required to be logged. Lastly, the *Water Level* super category covers the task of determining the amount of water in the pipe in terms how large a ratio of the pipe is under water. This is achieved by reporting the water level in steps of 10% from 0 to 100, with an uncertainty of ±5%. Unlike other defect observations that are annotated at specific occurrences, the water level (VA) is logged at the start and the end of the inspection, as well as when it transitions between severity levels, meaning the water level is known throughout the inspection. This is critical information during the inspection, as it gives an estimate of how much of the pipe is visible serving as a proxy uncertainty estimate for the inspection. Examples of each class in the supercategories are shown in Figure 2.2-2.5. More examples of the defect categories can be found in the supplementary materials of Paper B and examples of the water level can be found in Paper C.



(a) 0% **(b)** 20% **(c)** 40%

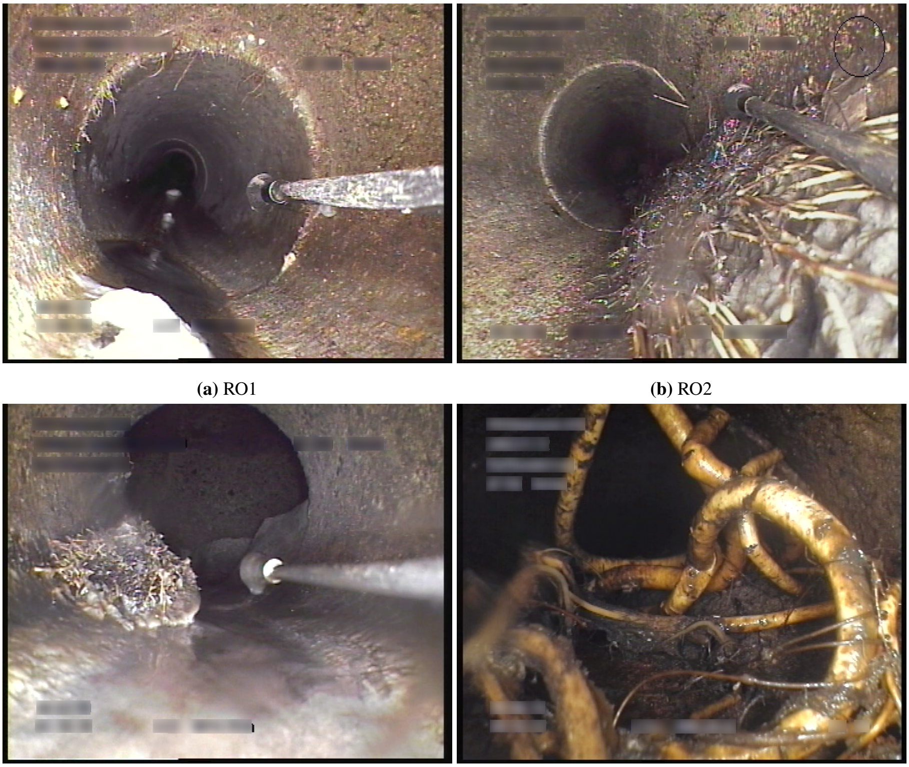
(d) 60% **(e)** 80% **(f)** 100%

Fig. 2.5: Water level severity. Examples of the water level category at a subset of the different severity levels. *Images from the Sewer-ML dataset [6].*

As mentioned earlier, each defect category has a set of predefined severity levels that the category can occur at, in order of increasing severity. An example of this can be seen in Figure 2.6, where the four severity levels of the root defect category are shown. Similarly, each category can have a set of types that can be used to further describe how the defect category has manifested itself inside the pipe. An example is shown for the infiltration category (IN) in Figure 2.7.

Between the 6th and 7th edition of the Fotomanual [5, 7], the two most recent editions, little has changed in these descriptions. The most noticeable difference is the change from categorizing the water level in steps of 10% and instead simply use four severity levels describing intervals ([0%-5%), [5%-15%), [15%-30%), and [30%-100%]) which are easier for the inspector to discriminate between.

Furthermore, for each observation there is a suite of extra metadata that should be logged. First, all observed defect categories has to be logged with exact timestamp as well as the distance traveled within the pipe. Secondly, each observation should be accompanied with a reference to where on the pipe wall the defect category is observed. This is achieved using a "clock reference", where the pipe is split into 12 segments, denoted hours, each covering 30 degrees compared to the pipe centerline. These clock references can be used to describe point and interval observations. An illustration of the clock reference system is shown in Figure 2.8. Thirdly, if there are several occurrences of the same defect category present within a pipe segment (1 meter), only the occurrence with the highest severity has to be logged. Lastly, the Danish standard dictates that a defect observation is denoted as "continuous" if it



(a) RO1

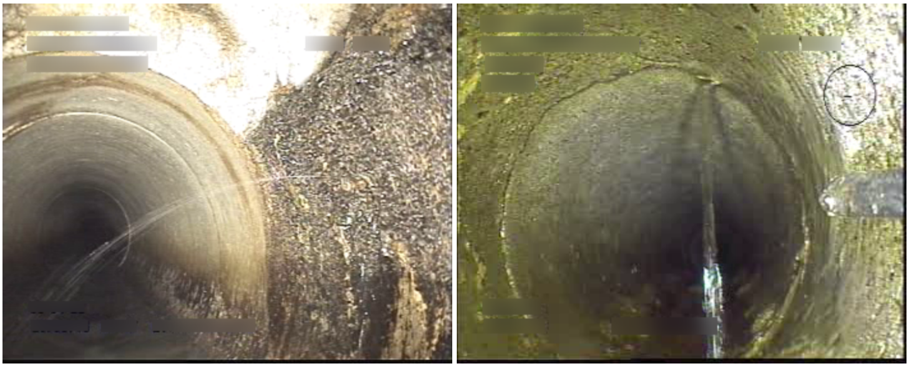
(b) RO2

(c) RO3

(d) RO4

Fig. 2.6: Root (RO) severity levels. Example of different severity levels for the same defect category, in this case the Roots defect category (RO), which contains four severity levels. *Images from the Sewer-ML dataset [6].*

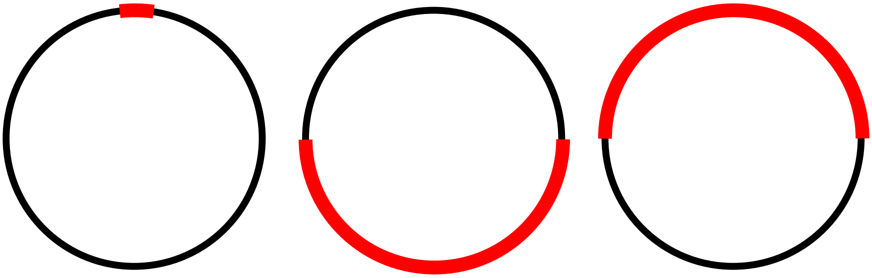
stretches over more than one meter. Normally, the Danish standard requires that each observation is logged as separate instances, except for the water level (VA) which is uniquely always continuous. However, this is not the case for continuous defects where only the start and end is required to be logged, as long as 80% of the occurrences of the defect category is at the originally observed severity level. If an occurrence of the same defect category but with different type indicators or a higher severity level, these have to be denoted separately in the document. While this notion of continuous defect can ease the job of the inspector and reduce tedious logging, it also introduces some uncertainty, as it is no longer known exactly where all defect occurrences are.



(a) IN3 Type R

(b) IN4 Type S

Fig. 2.7: Infiltration type indicators. Example of different type indicators for the same defect category. We highlight this with the infiltration defect category, which has two type indicators: **R** - the infiltration comes through the pipe wall, and **S** - the infiltration comes through a pipe joint. *Images from the Sewer-ML dataset [6].*



(a) Clock reference: 12

(b) Clock reference: 3-9

(c) Clock reference: 9-3

Fig. 2.8: Description of clock references. Examples of how the clock reference can be used to denote the extent of a defect.

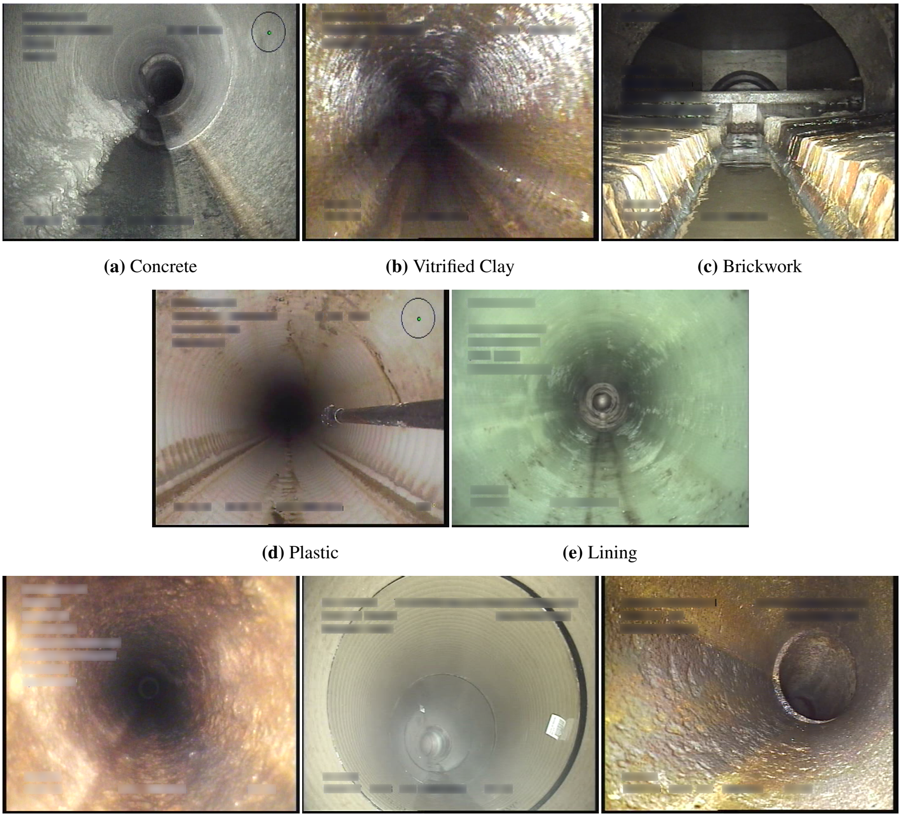
3.2 Pipe Properties

An equally essential part of the sewer inspection process is the logging of the pipe properties *i.e.* the material, shape, and diameter of the pipe. While it may not seem obvious the pipe properties directly affects how the defect categories are characterized and their severity classification. Specific for all pipe properties is the requirement of denoting how the property was determined. This can be by *e.g.* visually judging the properties or through consulting a construction plan.

The pipe material can be one of eight types, split into three overall categories, see Table 2.2. The two primary categories, Rigid and Flexible pipes, can be characterized by how the material reacts when deformed: Rigid pipes will have a set of cracks along the pipe wall, whereas Flexible pipes will simply deform. Extraordinarily, the iron material can be both rigid and flexible, depending on the specifics of the construction.

Table 2.2: Overview of the material classes in the Fotomanual. Three classes of materials are described in the Fotomanual: Rigid, Flexible, and Ambiguous. The pipe material has a direct influence on how defect categories are described and how severity levels manifest themselves.

Rigid	Flexible	Ambiguous
Concrete	Plastic	Iron
Vitrified Clay	Lining	Other
Brickwork		Unknown



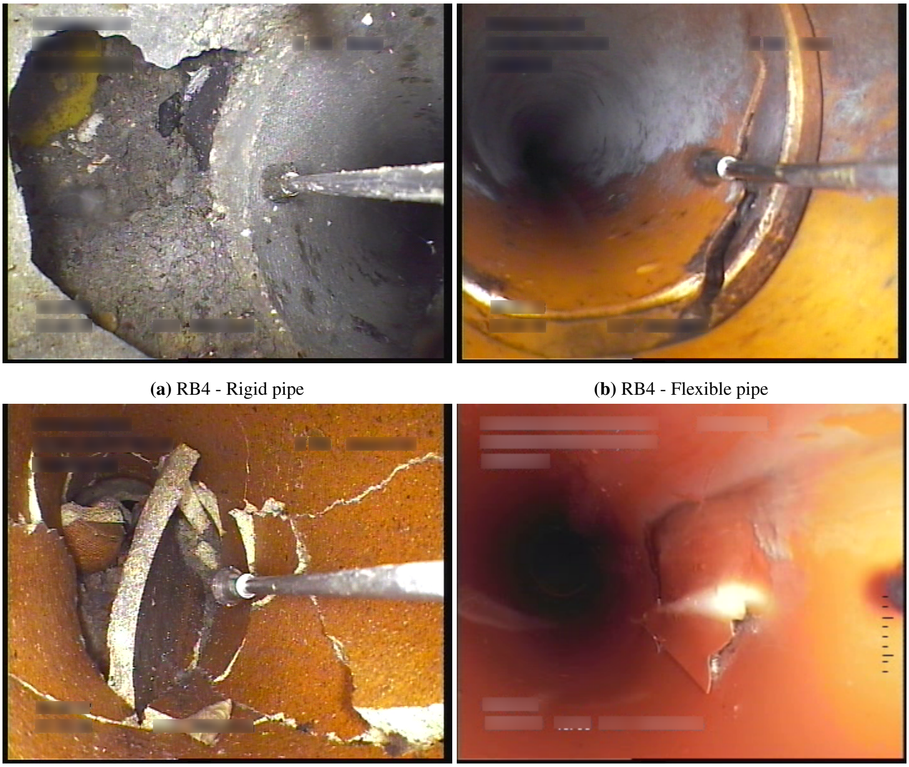
(a) Concrete **(b)** Vitrified Clay **(c)** Brickwork

(d) Plastic **(e)** Lining

(f) Iron **(g)** Other **(h)** Unknown

Fig. 2.9: Sewer pipe materials. Examples of the different sewer pipe materials according to the Fotomanual. *Images from the Sewer-ML dataset [6].*

Examples of all materials are shown in Figure 2.9 with more examples shown in the supplementary materials of Paper D.



(a) RB4 - Rigid pipe

(b) RB4 - Flexible pipe

(c) RB4 - Rigid pipe

(d) RB4 - Flexible pipe

Fig. 2.10: Effect of pipe material on severity level. Examples of how the defect severity level can be affected by the pipe material here highlighed using the crack, breaks, and collapses (RB) defect category at severity level 4. There is a clear difference in the visual presentation of the defect depending on the pipe material. *Images from the Sewer-ML dataset [6].*

Knowing the pipe material is essential in determining a subset of the defect categories and severity levels. This is best illustrated by the cracks, breaks and collapses (RB) defect category, where a RB4 (highest severity) can be a total collapse of the pipe if the material is rigid, whereas a single crack is enough when the material is flexible. This is illustrated with examples in Figure 2.10. Similarly, the deformation (DE) category is only used for flexible pipes, as a deformation of a rigid pipe automatically entails the presence of cracks and therefore instead classified as a RB3 or RB4.

Likewise, the pipe shape and diameter can affect how the defect category severity levels are characterized. For example, the displaced joint class (FS) the severity levels are based on the thickness of the pipe wall for rigid pipes, whereas for flexible pipes the severity levels are based on the pipe diameter. How the pipe diameter is determined is based on what kind of shape the pipe has out of six possible shapes: Circular, Conical, Egg, Eye, Rectangular, and Other. It is defined by the Fotomanual that the circular pipes are measured by the internal diameter, the conical pipes by the internal horizontal diameter, and all other shapes by the internal vertical diameter.

After a sewer inspection has been completed a single deterioration score can be calculated for the pipe ranging from perfect (0) to critical (10) condition [8]. This score is based on the overall metadata, the observed defect categories, their severity level, type indicators, and frequency.

References

[1] Danske TV-inspektionsfirmaers Kontrolordning (DTVK), "Regler og tekniske bestemmelser for danske tv-inspektionsfirmaers kontrolordning," 2018.

[2] J. B. Haurum and T. B. Moeslund, "A survey on image-based automation of cctv and sset sewer inspections," *Automation in Construction*, vol. 111, p. 103061, 2020.

[3] J. Dirksen, F. H. Clemens, H. Korving, F. Cherqui, P. L. Gauffre, T. Ertl, H. Plihal, K. Müller, and C. T. Snaterse, "The consistency of visual sewer inspection data," *Structure and Infrastructure Engineering*, vol. 9, no. 3, pp. 214–228, 2013.

[4] A. J. van der Steen, J. Dirksen, and F. H. Clemens, "Visual sewer inspection: detail of coding system versus data quality?" *Structure and Infrastructure Engineering*, vol. 10, no. 11, pp. 1385–1393, 2014.

[5] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 6th ed. Dansk Vand og Spildevandsforening (DANVA), 2010.

[6] J. B. Haurum and T. B. Moeslund, "Sewer-ml: A multi-label sewer defect classification dataset and benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[7] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 7th ed. Dansk Vand og Spildevandsforening (DANVA), 2015.

[8] ——, *Fotomanualen: Beregning af Fysisk Indeks ved TV-inspektion*, 1st ed. Dansk Vand og Spildevandsforening (DANVA), 2005.

Chapter 3

Foundations for Image-based Automation of Sewer Inspections

The sewerage infrastructure is a classic example of a part of modern society which laypersons neither thinks about unless it is malfunctioning, nor truly understand the sheer size of the infrastructure. For example, it is estimated that there is currently more than 1.28 million kilometers of public sewerage infrastructure and 800 thousand kilometers of private lateral connections in the United States (US) servicing nearly 240 million citizens, with an additional 56 million citizens expected to be connected by 2032 [1]. Meanwhile, the American Society of Civil Engineers (ASCE) gave the American sewerage infrastructure a D+ grade in 2017, signifying that major expansions and improvements are needed to properly deal with not just the current problems (*e.g.* an estimated 23000 – 75000 sewer overflow events a year) but also future demands of the infrastructure [1].

While the infrastructure can be expanded and improved, this is just a short term solution. The sewerage infrastructure is also dependent on regular inspections in order to ensure the structural and operational integrity of the sewerage system. This is impossible as the sewer inspections require professionally trained inspectors following inspection standards with a high level of complexity, as previously discussed in Chapter 2. This means that parts of the sewerage infrastructure may go for long time periods without being inspected, due to the lack of qualified personal.

To alleviate this problem, the field of image-based automation of sewer inspections is of great interest and importance. The aim of this research field is the development of computer vision-based algorithms which can fully or partially automate the sewer inspection process, such as classifying and detecting defects. This line of research is made viable by the large amount of annotated video data saved from historical inspections, and the great economic impact the increased inspection effectiveness brings. Herein lies the origins of a three decades old niche research branch of the computer vision field, with a potentially major societal, environmental, and sustainable impact.

These algorithms may be deployed in an assistive fashion when performing a sewer inspection, where the predictions are used to inform the sewer inspector about potential defects. This would help reduce some of the possible subjectiveness of the inspections that can occur when the inspector experiences fatigue. The algorithms may also be deployed on a robotic platform designed to inspect the pipes autonomously, enabling the parallel inspection of different sectors of the sewerage infrastructure on a large scale. However, these applications of automated sewer inspections focus on the on-site sewer inspections, while it is also viable to deploy the algorithms off-site. Automation of the off-site inspections enables faster inspections, as sewer inspectors can be instructed to simply not annotate any observations or limit the annotations to the most economical impactful defects. Instead, the sewer inspections would be performed off-site with the videos processed by the developed algorithms, allowing for efficient parallelization and decentralization of the inspection process. Similarly, a valuable application would be the analysis and re-analysis of historical sewer inspection videos. This would be needed if the annotation files are somehow lost and the inspection has to be repeated, or if an inspection is suspected to be incomplete.

In the following, we describe the evolution of the image-based automation of sewer inspection field in order to determine the fundamental underlying aspects and trends of the field, and how this has previously and currently affected the state and research direction of the field.

1 State-of-the-Art

Since the early 1990's computer vision has been used to recognize sewer defects from videos and images, in order to help automate the sewer inspection process [2]. Through the years there have been several clear trends in how the defect recognition problem has been approached.

While the CCTV sensors have been the dominating sensor throughout all three decades of the field, other sensors have been utilized. Specifically, sensors providing fish-eye views such as the IBAK PANORAMO tracktor [3] and the Sewer Scanner and Evaluation Technology (SSET) [4] sensor which combined fish-eye and standard CCTV video feeds to capture the entire pipe wall. However, these sensors have not managed to become a stable component of the sewer inspectors' equipment.

In the mid- to late-90's basic image processing methods were used to extract hand-crafted geometrical and intensity features such as connected edges and fitted ellipses from images (which were often binarized or converted to grayscale) and in turn used to classify, detect, and segment the defects [5–16]. These types of hand-crafted features were the predominantly used features for nearly three decades, with only few attempts to use statistical or frequency based features through method such as the discrete Haar wavelet transform, Gray-Level Co-Occurrence matrices, and statistical moments [7, 17–22], designed feature descriptors such as Scale-Invariant Feature Transform (SIFT), Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), and GIST [23–29], and most recently learned features [30–45].

On the contrary, the recognition methodology of choice has changed several times through the last three decades. Through the 90's the predominant recognition method was heuristic rules which were carefully designed based on a priori knowledge [5–8, 11, 13, 18, 46–53]. However, in the early and mid 00's these heuristic rules met competition as the data-driven shallow Neural Network (NN) methods became popular [9, 10, 15, 16, 30, 31, 39, 54–58]. These methods were primarily trained to classify potential defects based on the hand-crafted features [9, 10, 15], and only in few cases the full image [31, 39, 57]. This was the dominating recognition methodology until the late 10's, where other machine learning methods such as Support Vector Machine (SVM) and Random Forest (RF) based approaches were also adopted [17, 19–23, 25–28]. Interestingly, morphology-based recognition approaches were developed in parallel through this time period, which applied morphological operations together with heuristic rules such as ideal morphologies or size based classification [5, 9, 46, 59–61].

In 2018 the first large scale Convolutional Neural Networks (CNNs) were utilized for sewer defect recognition [37, 40, 43–45], signifying a paradigm shift within the field. Deep learning based approaches such as CNNs have since then been the most commonly used defect recognition methodology as well as feature description method [32–38, 40–42]. However, this has highlighted critical aspects in the field: the clear lack of open-source data and code, and a commonly agreed upon evaluation protocol.

Through the history of the field, everyone have used their own proprietary datasets of varying sizes, recorded in differing location, annotated according to different standards and using different subsets of classes, and evaluated using different metrics [2]. Combined with a culture of keeping software proprietary and not re-implementing others algorithms, these hindrances have resulted in it being near impossible to fairly compare algorithms across scientific work. This has forced the field to focus on case studies, where datasets are collected from a limited region and not expected to generalize across state or country borders. This is in stark contrast to the general computer vision field, where progress has been fostered by openly sharing code and data, as well as by deploying public leader boards where algorithms are evaluated identically [62, 63].

However, in recent years there have been attempts to break with the traditions of the field, and more fairly compare the scientific work. In 2021, a large scale multi-label sewer defect classification dataset called Sewer-ML was publicly released, consisting of 1.3 million annotated images [64]. Sewer-ML compared 12 state-of-the-art algorithms from the image-based sewer inspection and general multi-label classification fields using an evaluation protocol grounded in domain knowledge, highlighting clear shortcomings of the current algorithms. To further encourage research in the field, the dataset was publicly released together with all code, model weights, and a publicly available leader board. While there are no public datasets for the detection and segmentation tasks, there have still been observed a shift in how models are evaluated.

These recent advances indicate the beginning of a shift within the image-based automation of sewer inspection field towards a more open and fair research field, as limiting barriers such as access to data and code are removed.

2 Contributed Scientific Work

The work in this Ph.D. thesis has been focused on identifying and attempting to remove the main hindrances in the image-based automation of sewer inspections field. This has resulted in two papers published in both computer vision focused and inter-disciplinary research outlets, see appendix A–B.

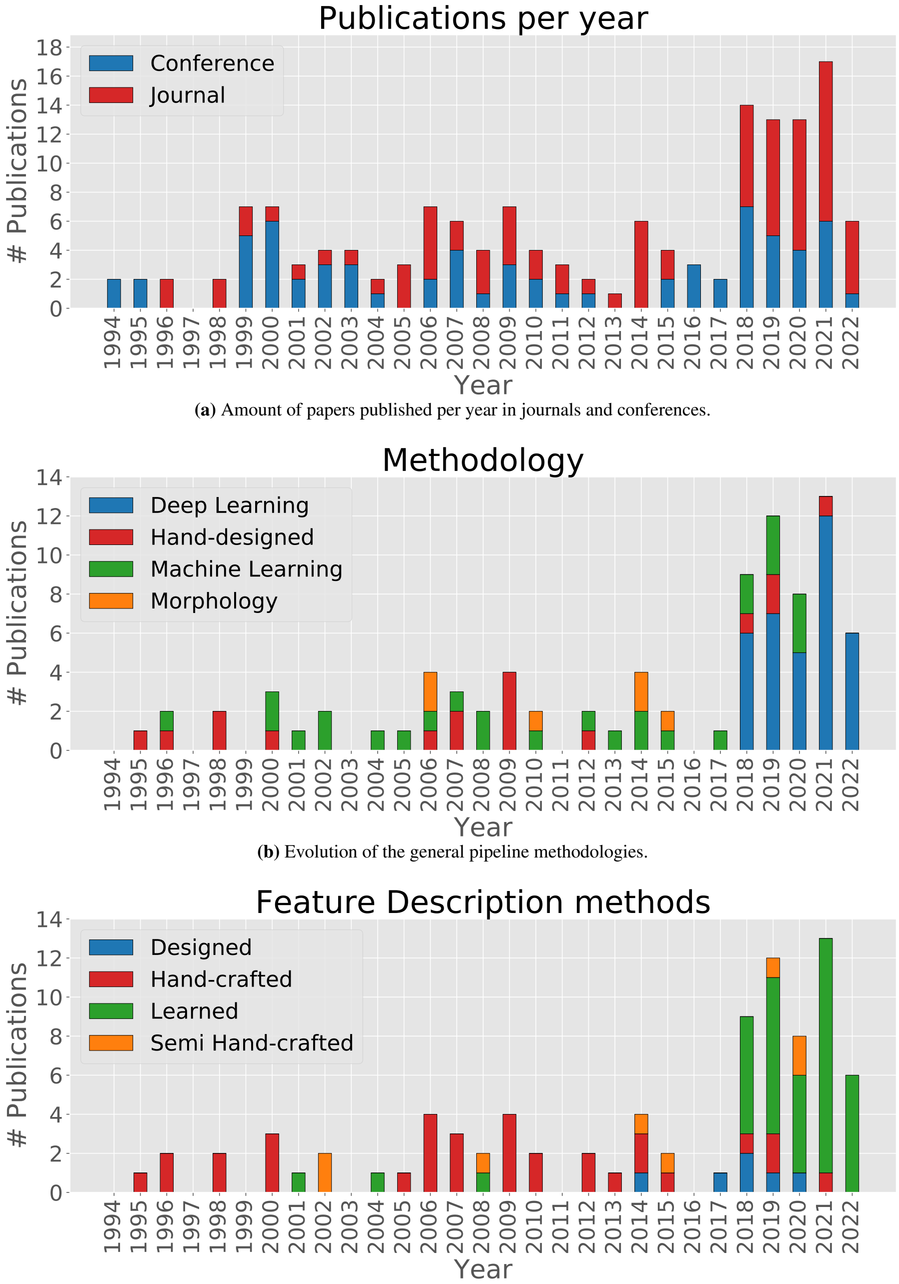
In Paper A we conducted the first survey dedicated to better understanding how image-based automation was used for sewer inspections. In total 113 papers published between 1994 and the start of 2020 were included in the survey. In Figure 3.1 we have updated key figures from Paper A with 39 papers released since the survey, to demonstrate the current state of the field[3], showing a clear continuation of the adaption of deep learning based approaches. Note that in Figure 3.1b – 3.1c datasets and methods developed over a series of papers are only represented by a single point, similar to the analysis conducted in Paper A.

The survey documented the entire image processing pipeline, comparing how the investigated methods addressed each step of the pipeline and how different methodologies came in and out of fashion. Through this analysis, it became clear that the image-based automation field was consistently lagging behind the computer vision field, while also developing specialized approaches using morphology-based classifiers. Similarly, the datasets and evaluation protocols were compared between all papers. Here it was made clear that there at the time was no consensus on what underlying data or inspection standard should be used, nor what kind of metric the model performance should be measured with. This directly meant that in principle the research findings are not directly comparable, due to significant differences in experimental design. All of these observations were found to be rooted in a prevalent tradition of not open-sourcing any parts of the research, leading to three major hindrances for the field:

- A lack of open-source code for easy reproduction and further development.
- A lack of a commonly agreed upon evaluation protocol.
- A lack of public datasets, readily available for all researchers.

In Paper B we aimed at breaking down the three hindrances which Paper A found were limiting the field. This was achieved by introducing the world's first publicly

[3] As of April 1, 2022



(a) Amount of papers published per year in journals and conferences.

(b) Evolution of the general pipeline methodologies.

(c) Evolution of the used feature description methods.

Fig. 3.1: The distribution of the papers published within the image-based automation of sewer inspection field, as well as the primary recognition methodology, and feature description methodology. *Adapted from [2], and updated with recent published works.*

Table 3.1: A comparison of datasets used for sewer defect classification. The following aspect are reported: Is the dataset publicly available (P), are the annotations multi-label (ML), the number of images with defects (DI), the number of images with normal pipes (NI), annotated classes (C), and the Class Imbalance (CI) for each dataset rounded to the nearest integer. Datasets are sorted by the number of images with defects. *Adapted from [64], and updated with data from recent published works.*

Dataset	**Year**	**P**	**ML**	**DI**	**NI**	**C**	**CI**
Gu *et al.* [65]	2021			965	8026	5	77
Ye *et al.* [66]	2019			1,045	0	7	13
Myrans *et al.* [27]	2018			2,260	0	13	102
Chen *et al.* [45]	2018			8,000	10,000	5	5
Li *et al.* [33]	2019			8,455	9,878	7	19
Kumar *et al.* [40]	2018			11,000	1,000	3	4
Ma *et al.* [67]	2021			14,451	0	4	2
Meijer *et al.* [35]	2019		✓	17,663	2,184,919	12	12,732
Situ *et al.* [68]	2021			20,290	0	4	2
Xie *et al.* [32]	2019			22,800	20,000	7	8
Hassan *et al.* [34]	2019			24,137	0	6	3
Dang *et al.* [69]	2021			24,452	13,934	8	6
Sewer-ML [64]	2021	✓	✓	609,479	690,722	17	123

available image-based sewer inspection dataset, Sewer-ML, based on 75,618 sewer inspection videos produced by professional sewer inspectors spanning over a 9 year period by the three largest Danish water utility companies and ASIR partners: HOFOR, VandCenter Syd, and Aarhus Vand. Even though all the data followed the same standard, it took nearly 5 months of work "cleaning" the data by *e.g.* fixing typos in the report files, tracking down missing or incorrectly named files, and correcting missing entries (such as a missing end annotation of a continuous defect observation). Furthermore, in order to extract the correct frame for an annotated defect, the timestamp in the inspection reports and the timestamp in the videos had to be synchronizes. This was achieved by annotating the on-screen timestamp at a predetermined frame in each video, which enabled us to synchronize the inspection videos and reports to ±1 second.

Using a set of heuristic rules grounded in the Danish sewer inspection process a total of 1,300,201 images were extracted with multi-label ground truth annotations. Examples of such heuristics were determining whether the inspection tracktor moved faster than the maximum allowed 0.25 m/s and exclude those periods from the extraction process, and only extracting "normal" images (*i.e.* images with no annotated defect categories) when the inspection tracktor is moving forward in the pipe. The on-screen information on all images were redacted through the use of a Faster-RCNN text detector, trained to detect text by using 23,044 images with manually annotated text boxes for training and validation. While all the information described in the Fotomanual was available for all images, it was decided to focus on the classification of the presence

Table 3.2: Benchmark results on Sewer-ML. Comparison of state-of-the-art methods from the sewer and general multi-label image classification domains. The metrics are presented as percentages, and the highest score in each column is denoted in bold. The "Sewer" and "General" identifiers indicate whether the method is from the sewer defect or multi-label classification domains, respectively. *Adapted from [64].*

	Model	**Validation**		**Test**	
		$F2_{CIW}$	**$F1_{Normal}$**	**$F2_{CIW}$**	**$F1_{Normal}$**
Sewer	Xie [32]	48.57	91.08	48.34	**90.62**
	Chen [45]	42.03	3.96	41.74	3.59
	Hassan [34]	13.14	0.00	12.94	0.00
	Myrans [27]	4.01	26.03	4.11	27.48
General	ResNet-101 [70]	53.26	79.55	53.21	78.57
	KSSNet [71]	54.42	80.60	54.55	79.29
	TResNet-M [72]	53.83	81.23	53.79	79.91
	TResNet-L [72]	54.63	81.22	54.75	79.88
	TResNet-XL [72]	54.42	81.81	54.24	80.42
	Benchmark [64]	**55.36**	**91.32**	**55.11**	**90.94**

of the different defects. Therefore, only the defect categories, excluding the water level defect category, were considered, resulting in 17 explicit classes and the implicit normal class. Uniquely this made Sewer-ML one of the largest datasets used across the image-based automation of sewer inspections field, while also truthfully representing the real-life class imbalance as no over- or under-sampling was performed. This is shown in Table 3.1. The Sewer-ML dataset enabled the fair comparison state-of-the-art algorithms from both the sewer inspection and multi-label classification domains. Six representative algorithms were chosen from each of the domains, implemented in the same open-source codebase, and trained from scratch. From the sewer domain this included an ensemble of binary classifiers [40], two-stage classifiers consisting of a small binary classifier and larger multi-label classifier [27, 32, 45], and end-to-end classifiers [34, 35]. In contrast, the general multi-label image classification domain only consisted of end-to-end classifiers. Two state-of-the-art graph-based methods, MLGCN [73] and KSSNet [71]. were chosen, as well as the common ResNet-101 backbone [70] and recent TResNet backbones [72].

The model performances were measured using the F1-score for the implicit normal class, $F1_{Normal}$ (Eq. 3.1), and a weighted F2-score, $F2_{CIW}$ (Eq. 3.2), where each class is weighted by their *class importance weight* (CIW), which is based on the weighting used in the Danish standard to determine the pipe deterioration score [74]. The F2-score was chosen for evaluating defect categories as the F2-score weights the recall of the classifier higher than the precision of the classifier. This directly reflects that missing a defect can result in a higher economic impact as a faulty pipe goes unnoticed, whereas falsely predicting a defect can quickly be corrected by humans verifying the detected defects before initiating a restoration project.

$$\mathrm{F1}_{\mathrm{Normal}} = 2\frac{\mathrm{Prc}_{\mathrm{Normal}} \cdot \mathrm{Rcll}_{\mathrm{Normal}}}{\mathrm{Prc}_{\mathrm{Normal}} + \mathrm{Rcll}_{\mathrm{Normal}}} \quad (3.1) \qquad \mathrm{F2}_{\mathrm{CIW}} = \frac{\sum_{c=1}^{C} \mathrm{F2}_c \cdot \mathrm{CIW}_c}{\sum_{c=1}^{C} \mathrm{CIW}_c}, \quad (3.2)$$

where $\mathrm{Prc}_{\mathrm{Normal}}$ and $\mathrm{Rcll}_{\mathrm{Normal}}$ are the precision and recall scores for the normal class, CIW_c and $\mathrm{F2}_c$ are the CIW and F2-score for class c, respectively, and C is the number of annotated classes.

Through this analysis, we found that none of the current state-of-the-art methods could achieve a $\mathrm{F1}_{\mathrm{Normal}}$ score higher than 90.62% or $\mathrm{F2}_{\mathrm{CIW}}$ score higher than 54.75%, see Table 3.2, and that three of the tested methods diverged during training. By creating a two-stage approach by combining the first stage of the two-stage classifier of Xie *et al.* [32] and the TResNet-L multi-label classifier [72], denoted as the *Benchmark* algorithm, a state-of-the-art performance $\mathrm{F1}_{\mathrm{Normal}}$ score of 90.94% and $\mathrm{F2}_{\mathrm{CIW}}$ score of 55.11% was achieved. When looking at the per-class performance of the Benchmark algorithm, it became clear that the classes with a high F2-score displayed low intra-class variance and high inter-class variance, while the opposite was true for classes with low F2-scores. This strongly shows that the sewer defect classification task is far from solved as some have previously claimed, and that further attention to the task is needed. In order to enable this change, a public online leader board was established[4].

3 Contributions

We have investigated and highlighted the fundamental trends and hindrances of the image-based automation of sewer inspections field by surveying the last three decades of published work. In an attempt to remove some of these hindrances, we have released the world's first public sewer defect classification dataset, benchmarked 12 relevant algorithms using domain influenced metrics, and found that the defect classification task is far from solved. Our main contributions within the fundamentals aspects of the image-based automation of sewer inspections field are thus:

- A systematic overview of the field, documenting research trends through three decades. This led to the identification of three major hindrances in the field: (1) the lack of publicly available datasets, (2) the lack of a commonly agreed upon evaluation protocol, and (3) the lack of open-sourced code.
- The Sewer-ML dataset, the world's first publicly available image-based sewer inspection dataset framed as a multi-label classification task, containing 1.3 million images from real sewer inspections annotated according to the Fotomanual.
- Two evaluation metrics, $\mathrm{F1}_{\mathrm{Normal}}$ and $\mathrm{F2}_{\mathrm{CIW}}$, which directly incorporates domain knowledge into the evaluation protocol by considering the economic consequence of defect categories and sensitivity of the classifier.

[4] https://competitions.codalab.org/competitions/32705

- A comprehensive comparison of twelve state-of-the-art methods from the image-based sewer inspection domain and the multi-label classification label, and a proposed two-stage benchmark algorithm achieving a $\text{F1}_{\text{Normal}}$ score of 90.94% and F2_{CIW} score of 55.11%.
- A conscientious effort to open-source all code in order to facilitate scientific reproduction and act as a stepping stone for future work, in the hope of spearheading a more transparent research culture in the automated sewer inspection field.

References

[1] American Society of Civil Engineers, “2017 infrastructure report card - wastewater,” 2017, accessed: 20/3-2022. [Online]. Available: https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf

[2] J. B. Haurum and T. B. Moeslund, “A survey on image-based automation of cctv and sset sewer inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[3] IBAK Helmut Hunger GmbH & Co. KG, “Panoramo 4k system,” accessed: 21/3-2022. [Online]. Available: https://www.ibak.de/en/produkte/ibak_show/frontenddetail/product/panoramo-4k-system/

[4] M. J. Chae, T. Iseley, and D. M. Abraham, “Computerized sewer pipe condition assessment,” in *New Pipeline Technologies, Security, and Safety*, 2003, pp. 477–493.

[5] S. K. Sinha and P. W. Fieguth, “Morphological segmentation and classification of underground pipe images,” *Machine Vision and Applications*, vol. 17, no. 1, p. 21, Jan 2006.

[6] M. R. Halfawy and J. Hengmeechai, “Efficient algorithm for crack detection in sewer images from closed-circuit television inspections,” *Journal of Infrastructure Systems*, vol. 20, no. 2, p. 04013014, 2014.

[7] A. Hawari, M. Alamin, F. Alkadour, M. Elmasry, and T. Zayed, “Automated defect detection tool for closed circuit television (cctv) inspected sewer pipelines,” *Automation in Construction*, vol. 89, pp. 99 – 109, 2018.

[8] G. Heo, J. Jeon, and B. Son, “Crack automatic detection of cctv video of sewer inspection with low resolution,” *KSCE Journal of Civil Engineering*, vol. 23, no. 3, pp. 1219–1227, Mar 2019.

[9] S. K. Sinha and P. W. Fieguth, “Neuro-fuzzy network for the classification of buried pipe defects,” *Automation in Construction*, vol. 15, no. 1, pp. 73 – 83, 2006.

[10] T. Shehab and O. Moselhi, “Automated detection and classification of infiltration in sewer pipes,” *Journal of Infrastructure Systems*, vol. 11, no. 3, pp. 165–171, 2005.

[11] A. Ahrary, Y. Kawamura, and M. Ishikawa, “An automated intelligent fault detection system for inspection of sewer pipes,” *IEEJ Transactions on Electronics, Information and Systems*, vol. 127, no. 6, pp. 943–950, 2007.

[12] W. Guo, L. Soibelman, and J. Garrett, "Automated defect detection for sewer pipeline inspection and condition assessment," *Automation in Construction*, vol. 18, no. 5, pp. 587 – 596, 2009.

[13] W. Guo, L. Soibelman, and J. H. Garrett, "Visual pattern recognition supporting defect reporting and condition assessment of wastewater collection systems," *Journal of Computing in Civil Engineering*, vol. 23, no. 3, pp. 160–169, 2009.

[14] A. Chaki and T. Chattopadhyay, "An intelligent fuzzy multifactor based decision support system for crack detection of underground sewer pipelines," in *2010 10th International Conference on Intelligent Systems Design and Applications*, Nov 2010, pp. 1471–1475.

[15] O. Moselhi and T. Shehab-Eldeen, "Classification of defects in sewer pipes using neural networks," *Journal of Infrastructure Systems*, vol. 6, no. 3, pp. 97–104, 2000.

[16] I. Khalifa, A. E. Aboutabl, and G. S. A. Barakat, "A new image-based model for predicting cracks in sewer pipes," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 12, pp. 65–71, 2013.

[17] M. Browne, M. Dorn, R. Ouellette, T. Christaller, and S. Shiry, "Wavelet entropy-based feature extraction for crack detection in sewer pipes," in *6th International Conference on Mechatronics Technology, Kitakyushu, Japan*, 2002, pp. 202–206.

[18] K. Müller and B. Fischer, "Objective condition assessment of sewer systems," in *2nd Leading Edge Conference on Strategic Asset Management*, 2007, pp. 641–652.

[19] M.-D. Yang and T.-C. Su, "Automated diagnosis of sewer pipe defects based on machine learning approaches," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1327 – 1337, 2008.

[20] J. Mashford, M. Rahilly, B. Lane, D. Marney, and S. Burn, "Edge detection in pipe images using classification of haar wavelet transforms," *Applied Artificial Intelligence*, vol. 28, no. 7, pp. 675–689, 2014.

[21] W. Wu, Z. Liu, and Y. He, "Classification of defects with ensemble methods in the automated visual inspection of sewer pipes," *Pattern Analysis and Applications*, vol. 18, no. 2, pp. 263–276, May 2015.

[22] X. Ye, J. Zuo, R. Li, Y. Wang, L. Gan, Z. Yu, and X. Hu, "Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern chinese city," *Frontiers of Environmental Science & Engineering*, vol. 13, no. 2, p. 17, Jan 2019.

[23] C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, "A vision-based system for internal pipeline inspection," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3289–3299, June 2019.

[24] S. Moradi, T. Zayed, and F. Golkhoo, "Automated sewer pipeline inspection using computer vision techniques," in *Pipelines 2018*, 2018, pp. 582–587.

[25] S. Moradi and T. Zayed, "Real-time defect detection in sewer closed circuit television inspection videos," in *Pipelines 2017*, 2017, pp. 295–307.

[26] J. Myrans, R. Everson, and Z. Kapelan, "Automated detection of faults in sewers using cctv image sequences," *Automation in Construction*, vol. 95, pp. 64 – 71, 2018.

[27] ——, "Automated detection of fault types in cctv sewer surveys," *Journal of Hydroinformatics*, vol. 21, no. 1, pp. 153–163, Oct 2018.

[28] M. R. Halfawy and J. Hengmeechai, “Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine,” *Automation in Construction*, vol. 38, pp. 1 – 13, 2014.

[29] W. Guo, L. Soibelman, and J. J. H. Garrett, “Automated defect detection in urban wastewater pipes using invariant features found in video images,” in *Building a Sustainable Future*, 2009, pp. 1194–1203.

[30] D. Bairaktaris, V. Delis, C. Emmanouilidis, S. Frondistou-Yannas, K. Gratsias, V. Kallidromitis, and N. Rerras, “Decision-support system for the rehabilitation of deteriorating sewers,” *Journal of Performance of Constructed Facilities*, vol. 21, no. 3, pp. 240–248, 2007.

[31] M. Browne, S. S. Ghidary, and N. M. Mayer, “Convolutional neural networks for image processing with applications in mobile robotics,” in *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks*, B. Prasad and S. R. M. Prasanna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 327–349.

[32] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, “Automatic detection and classification of sewer defects via hierarchical deep learning,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2019.

[33] D. Li, A. Cong, and S. Guo, “Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification,” *Automation in Construction*, vol. 101, pp. 199 – 208, 2019.

[34] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, “Underground sewer pipe condition assessment based on convolutional neural networks,” *Automation in Construction*, vol. 106, p. 102849, 2019.

[35] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, “A defect classification methodology for sewer image sets with convolutional neural networks,” *Automation in Construction*, vol. 104, pp. 281 – 298, 2019.

[36] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, “A deep learning-based framework for an automated defect detection system for sewer pipes,” *Automation in Construction*, vol. 109, p. 102967, 2020.

[37] J. C. Cheng and M. Wang, “Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,” *Automation in Construction*, vol. 95, pp. 155 – 171, 2018.

[38] M. Wang and J. C. P. Cheng, “A unified convolutional neural network integrated with conditional random field for pipe defect segmentation,” *Computer-Aided Civil and Infrastructure Engineering*, pp. 1–15, 2019.

[39] M. J. Chae and D. M. Abraham, “Neuro-fuzzy approaches for sanitary sewer pipeline condition assessment,” *Journal of Computing in Civil Engineering*, vol. 15, no. 1, pp. 4–14, 2001.

[40] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, “Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks,” *Automation in Construction*, vol. 91, pp. 273 – 283, 2018.

[41] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, “Deep learning-based automated detection of sewer defects in cctv videos,” *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[42] S. S. Kumar and D. M. Abraham, "A deep learning based automated structural defect detection system for sewer pipelines," in *Computing in Civil Engineering 2019*, 2019, pp. 226–233.

[43] J. Kunzel, T. Werner, P. Eisert, and J. Waschnewski, "Automatic analysis of sewer pipes based on unrolled monocular fisheye images," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018, pp. 2019–2027.

[44] R. Tennakoon., R. Hoseinnezhad., H. Tran., and A. Bab-Hadiashar., "Visual inspection of storm-water pipe systems using deep convolutional neural networks," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, INSTICC. SciTePress, 2018, pp. 135–140.

[45] K. Chen, H. Hu, C. Chen, L. Chen, and C. He, "An intelligent sewer defect detection method based on convolutional neural network," in *2018 IEEE International Conference on Information and Automation (ICIA)*, Aug 2018, pp. 1301–1306.

[46] S. Iyer and S. K. Sinha, "Segmentation of pipe images for crack detection in buried sewers," *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 6, pp. 395–410, 2006.

[47] X. Pan, T. Ellis, and T. Clarke, "Robust tracking of circular features." in *Proceedings of the British Machine Vision Conference*. BMVA Press, 1995, pp. 55.1–55.10.

[48] K. Xu, A. Luxmoore, and T. Davies, "Sewer pipe deformation assessment by image analysis of video surveys," *Pattern Recognition*, vol. 31, no. 2, pp. 169 – 180, 1998.

[49] S. Broadhurst, G. Cockerham, N. Taylor, and T. Pridmore, "Automatic task modelling for sewer studies," *Automation in Construction*, vol. 5, no. 1, pp. 61 – 71, 1996, 12th ISARC.

[50] N. Taylor, T. Pridmore, and S. Fu, "Automatic visual detection of lateral junctions in sewers." *Proceedings of the Institution of Civil Engineers - Water, Maritime and Energy*, vol. 130, no. 2, pp. 56–69, 1998.

[51] P. Swarnalatha, M. Kota, N. R. Resu, and G. Srivasanth, "Automated assessment tool for the depth of pipe deterioration," in *2009 IEEE International Advance Computing Conference*, March 2009, pp. 721–724.

[52] W. Xue-Fei and B. Hua, "Automated assessment of buried pipeline defects by image processing," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, Nov 2009, pp. 583–587.

[53] P. Huynh, R. Ross, A. Martchenko, and J. Devlin, "Dou-edge evaluation algorithm for automatic thin crack detection in pipelines," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Oct 2015, pp. 191–196.

[54] J. R. del Solar and M. Köppen, "Sewage pipe image segmentation using a neural based architecture," *Pattern Recognition Letters*, vol. 17, no. 4, pp. 363 – 368, 1996, neural Networks for Computer Vision Applications.

[55] L. Paletta and E. Rome, "Learning fusion strategies for visual object detection," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 2, Oct 2000, pp. 1446–1452 vol.2.

[56] M. Browne, S. Shiry, M. Dorn, and R. Ouellette, "Visual feature extraction via pca-based parameterization of wavelet density functions," in *International Symposium on Robots and Automation*, 2002, pp. 398–402.

[57] R. Oullette, M. Browne, and K. Hirasawa, “Genetic algorithm optimization of a convolutional neural network for autonomous crack detection,” in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, June 2004, pp. 516–521 Vol.1.

[58] H. Ganegedara, D. Alahakoon, J. Mashford, A. Paplinski, K. Müller, and T. M. Deserno, “Self organising map based region of interest labelling for automated defect identification in large sewer pipe image collections,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, June 2012, pp. 1–8.

[59] M.-D. Yang and T.-C. Su, “Segmenting ideal morphologies of sewer pipe defects on cctv images for automated diagnosis,” *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3562 – 3573, 2009.

[60] T.-C. Su, M.-D. Yang, T.-C. Wu, and J.-Y. Lin, “Morphological segmentation based on edge detection for sewer pipe defects on cctv images,” *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 094 – 13 114, 2011.

[61] T.-C. Su and M.-D. Yang, “Application of morphological segmentation to leaking defect detection in sewer pipelines,” *Sensors*, vol. 14, no. 5, pp. 8686–8704, 2014.

[62] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[63] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[64] J. B. Haurum and T. B. Moeslund, “Sewer-ml: A multi-label sewer defect classification dataset and benchmark,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[65] Y. Gu, W. Tu, Q. Li, T. Zhao, D. Zhao, S. Zhu, and J. Zhu, “Collaboratively inspect large-area sewer pipe networks using pipe robotic capsules,” in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 211–220.

[66] X. Ye, J. Zuo, R. Li, Y. Wang, L. Gan, Z. Yu, and X. Hu, “Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern chinese city,” *Frontiers of Environmental Science & Engineering*, vol. 13, no. 2, Jan. 2019.

[67] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, and J. Dong, “A multi-defect detection system for sewer pipelines based on stylegan-sdm and fusion cnn,” *Construction and Building Materials*, vol. 312, p. 125385, 2021.

[68] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, “Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known cnn classifier,” *IEEE Access*, vol. 9, pp. 59 498–59 507, 2021.

[69] L. M. Dang, S. Kyeong, Y. Li, H. Wang, T. N. Nguyen, and H. Moon, “Deep learning-based sewer defect classification for highly imbalanced dataset,” *Computers & Industrial Engineering*, vol. 161, p. 107630, 2021.

[70] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, pp. 770–778.

[71] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, “Multi-label classification with label graph superimposing,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 265–12 272, Apr. 2020. [Online]. Available: https://doi.org/10.1609/aaai.v34i07.6909

[72] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, “Tresnet: High performance gpu-dedicated architecture,” *CoRR*, 2020.

[73] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5172–5181.

[74] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: Beregning af Fysisk Indeks ved TV-inspektion*, 1st ed. Dansk Vand og Spildevandsforening (DANVA), 2005.

Chapter 4

Advancing Image-based Automation of Sewer Inspections

As described in Chapter 3, the image-based automation of sewer inspection field has a three decades old history in parallel with the general computer vision domain. However, the trends and advances made within the general computer vision domain have not always transferred over to this niche research branch due to culture differences between the computer science and civil engineering area and major commercial interests. This has caused the automated sewer inspection field to continuously lag behind the general compute vision field by approximately five years.

However, with the adoption of deep learning based approaches and general guidelines for fair comparison of algorithms, the field has begun advancing at a faster pace than previously experienced. This has led to advances across all areas of defect recognition and advances towards automatically completing larger parts of the sewer inspection process.

1 State-of-the-Art

With the image-based automation field shifting from the classical methodologies to fully embracing deep learning methods [1], the main focus of the field has been on developing and investigating the usability of the data-driven neural networks for classification, detection, and segmentation.

Within the sewer defect detection area this has been achieved by applying networks from the general computer vision field on domain specific data from the image-based sewer inspection field. This has led to the widespread adoption of object detectors such as Faster R-CNN, Single Shot Detector (SSD), and different variations of the

You Only Look Once network (YOLO) [2–8]. Most recently, there has been attempts to use more modern computer vision techniques. Siu *et al.* [9] proposed a synthetic data generator for defect detection, where textured 3D pipe segment models with randomly chosen defects (initially restricted to cracks), water level, and pipe material. In order to reduce the domain gap between real and synthetic data, a style transfer network was used to transfer the style of the real sewer inspection images onto the synthetic data. A Faster-RCNN was used to benchmark the effect of the synthetic data, with the addition of a contrastive loss to the output of the region-of-interest pooling step in the Faster-RCNN architecture. Dang *et al.* proposed adapting the transformer-based DETR architecture [10] to the sewer domain, in a model denoted DefecTR [11]. DefecTR works by adjusting the training process of DETR with a new loss and smaller adjustments, using a Sewer-ML [12] pre-trained backbone, and preprocessing the images to reduce haze and enhance the contrast of the images, using the Gated Context Aggregation Network (GCANet) [13] and Dynamic Histogram Equalization (DHE) [14], respectively. Using the area of the averaged attention maps, DefecTR can determine the Zone of Influence (ZoI) of a defect, which in combination with the predicted class made DefecTR able to predict the severity of the class according to the North American PACP standard.

Similarly, within the sewer defect segmentation area progress has been made by applying and adapting general segmentation approaches to the sewer inspection domain. For semantic segmentation this is demonstrated by the PipeUNet [15], which adds an attention based shortcut to the common U-Net architecture, and the DialSeg-CRF [16], which adds a Conditional Random Field (CRF) formulated as a Recurrent Neural Network (RNN) to the backbone network. For instance segmentation, the Pipe-SOLO network [17] has been introduced where the SOLOv2 architecture [18] is modified by replacing the common Feature Pyramid Network (FPN) [19] head with an enhanced bi-directional feature pyramid (EBiFPN) head, while also dehazing the images using the GCANet.

Common for both the detection and segmentation areas is the increased attention to fair comparison between models. In recent studies, comparisons between different architectures have been thoroughly studied on the utilized datasets, to demonstrate the effectiveness of the proposed methods.

In contrast the sewer defect classification area has seen comparatively less progress. Except for the recent Multi-Scale Hybrid Vision Transformer (MSHViT) [20] that achieves state-of-the-art on Sewer-ML by building upon the Hybrid Vision Transformer (HViT) [21] concept of combining CNN and Vision Transformers (ViTs), none of the newer classification networks such as EfficientNet [22] or ViTs have been adopted by the field. Instead, the image-based sewer defect classification field has focused on the development and deployment of CNNs on embedded devices [23, 24], and producing possible solutions to the data collection problem. One such approach utilized a subset of the Sewer-ML dataset for the weakly supervised object detection problem based on an attention-based mechanism, allowing for efficient use of the Sewer-ML dataset without expensive manual localization labeling [25]. Dang *et al.* [26] similarly investigated the localization capability of classifier networks using explainable AI techniques such as Class Activation Maps (CAM) [27] and layer activation visualization, while classifying imbalanced data with an ensemble of a VGG-19 network [28] and gradient boosting techniques with VGG-19 features.

Alternatively, a possible solution to the data collection problem has focused on using variations of the StyleGANv2 network [29] to generate synthetic high-quality data for the classification problem [30, 31]. These works have focused on using small datasets on which a pre-trained StyleGANV2 is fine-tuned, and thereafter fine-tuning classification networks using just synthetic data [30] or a combination of real and synthetic data [31]. Through this approach it was demonstrated that it is possible to achieve a high classification performance while utilizing synthetic sewer data. A common theme for these GAN based approaches has been the use of very small datasets with very few defect categories.

Recently, the field has also started moving towards developing approaches where the entire sewer inspection process is taken into consideration, and not just the defect recognition task.

This has led to the first application of neural networks for sewer defect tracking [4], where the defects are first detected using a Faster-RCNN, and the detected defect area is passed through a network trained in a metric learning fashion in order to produce distinct features per defect. In order to associate detection with tracks, an association cost is calculated for each combination of tracks and detected defect, based on appearance, motion, and defect class distance. The association cost between all defects and tracks is then minimized using the Hungarian algorithm.

In parallel, the image-based automation field has also started investigating how to classify the sewer pipe properties, and not just the sewer defects. Ji *et al.* [32, 33] proposed a segmentation based approach for determining the water level and pipe elevation in a sewer pipe, by using a pretrained DeepLabv3 network and leveraging that the shape of the pipe is known. This approach was verified in a case study where data from a single pipe was collected with a few hundred images annotated, and found to work at a human level as well as beat a classical image processing method.

Similarly, the Sewer-ML has been used to investigate classification of water level according to the Fotomanual standard. First a study was conducted by Haurum *et al.* [34] on a subset of the Sewer-ML dataset, in order to determine the effectiveness of decision tree-based and CNN-based methods that have been shown to work well on sewer defect classification. Through this analysis it was verified that the revised clustered severity levels introduced in the 7th edition of the Fotomanual leads to a better classification rate than the previously used severity levels in the 6th edition Fotomanual. Rius *et al.* [35] continued work on this task using the full Sewer-ML dataset, investigating the effect of self-supervised learning with an array of Variational Autoencoders (VAEs) [36], and training an MLP classifier or regressor on the latent embeddings. Using the original labels they did not outperform the initial baseline from the prior work, however, after revising the labels the proposed methodology showed significant improvements.

This line of water level classification was continued in the first application of multi-task classification (MTC) in the sewer domain for simultaneously predicting sewer defect classes as well as water level, pipe shape, and pipe material [37]. Using the Sewer-ML dataset it was shown that the general MTC approach where the tasks share a common encoder and separate class decoders improved performance considerably for all tasks, with the shape and material tasks improving by 35 and 15 percentage points, respectively. Haurum *et al.* [37] proposed the Cross-Task Graph Neural Network (CT-GNN) to further improve this performance by encoding co-occurrence information of the classes within and between tasks into a graph, which the CT-GNN utilizes to refine class specific features across all tasks in the decoder stage of the MTC network. By including the cross-task information, the performance increased further while introducing 50 times fewer parameters compared to prior MTC approaches. This work shows the advantages of using the MTC paradigm for training a single network that is multi-purpose, perfect for online inspections where predictions need to occur rapidly.

Similarly, Wang *et al.* [5] proposed a framework to classify Operation and Maintenance (O&M) defects and the severity following the North American PACP standard[5]. First, the pipe cross section is estimated by fitting an ellipse to the pipe joint, with the implicit assumption that the pipe is circular. This is achieved using an traditional image processing approach, where the image is converted to grayscale and edges are extracted using the Canny edge detector. As images often contain text, these text regions are detected using the Maximal Stable Extremal Regions (MSER) method [38]. MSER has been used previously to help auto-generate inspection reports for offline processing of videos, by detecting and analyzing the text in the video frames [26, 39–41]. However, in this case MSER is used to detect text regions such that the text can be subtracted from the edge map produced by the Canny edge detector. Lastly, an ellipse is fitted to the remaining edges using a least squares approach. The defects are extracted using a Faster R-CNN object detector, and if an O&M defect is detected, the DilaSeg-CRF semantic segmentation model is applied in order to get pixel level localization of the defect. This builds upon their previous work within the field [2, 16]. With the defect segmented it is possible to determine the area of the defect, and thereby the ratio of the defect compared to the cross section of the pipe. Using this ratio it is possible to determine the severity of the O&M defects. Together with the DefecTR paper, these studies are interesting initial investigations into determining not just the defect category but also the severity levels. This has already led to follow up work by Zhou *et al.* [42], who compare several semantic segmentation networks and simply compares the defect area with the image area, and not the area of the cross section used by Wang *et al.* While this increased interest in defect severity classification is welcome, the current approaches do suffer in different ways. Both the Wang *et al.* and Zhou *et al.* studies are hampered by relatively small dataset sizes, whereas the DefecTR approach produces a

[5] O&M defects are equivalent to defects in the Operational Condition super category when following the Danish Fotomanual

cruder approximation of the severity level using ZoI, which is only inspired by and not based on the PACP standard. Furthermore, all three studies only work on a small set of classes from the sewer inspection standard, such as tree roots, cracks, deposits, and displaced joints.

2 Contributed Scientific Work

The work in this Ph.D. thesis has been focused on building on top of the fundamental contributions documented in Chapter 3 and further advance the state-of-the-art in the field of image-based automation of sewer inspection. This has resulted in three papers published in both computer vision focused and inter-disciplinary research outlets, see appendix C–E.

In Paper C a subset of 5511 inspection videos from the Sewer-ML dataset was used to investigate the effectiveness of using computer vision to infer the water level in images from real sewer inspection. Up to 1200 images were sampled for all 11 severity levels, annotated as per the 6th edition Fotomanual [43], and split into training, validation and test splits. As the automated sewer inspection domain had shown very little interest in the water level task previously, there were no obvious methods to compare with. Therefore, we compared the Random Forest [44] and Extra Trees [45] decision tree-based approaches with the AlexNet [46] and ResNet-{18, 34, 50} [47] CNN-based approaches commonly used in the defect classification task. We perform a thorough hyperparameter search for the tree-based approaches, investigating the effect of the number of trees used, the maximum depth of the trees, and number of features used in each decision point. For the CNN approaches we compared the performance when either training from scratch or fine-tuning an ImageNet pretrained model. Furthermore, we investigated the effect of how the data was labeled and how the task was presented, through three different label settings: An 11-way classification task, trained either as a classification task or regression task, denoted Class10 and Ref2Class10, respectively, and a 4-way classification task, denoted Class15, transforming the labels into the equivalent labels from the 7th edition of the Fotomanual. All models were evaluated using the micro-F1 and macro-F1 scores, see Eq. 4.3-4.4, as the dataset was not perfectly balanced. The micro-F1 score was chosen in order to capture the global performance where the specific class performance is not considered, leading to a higher sensitivity to majority classes. In contrast, the macro-F1 score is computed as the arithmetic mean of per-class F1-scores, causing the macro-F1 score to be more sensitive to minority classes, as all classes are weighted equally.

$$\text{micro-Prc} = \frac{\sum_{c=1}^{C} \text{TP}_c}{\sum_{c=1}^{C} \text{TP}_c + \text{FP}_c} \quad (4.1) \qquad \text{micro-Rcll} = \frac{\sum_{c=1}^{C} \text{TP}_c}{\sum_{c=1}^{C} \text{TP}_c + \text{FN}_c} \quad (4.2)$$

Table 4.1: Water level classification under varying label settings. Comparison of results when the water level classification task is posed as a 11-way classification following the 6th edition Fotomanual from 2010 (Class10), a regression problem following the 6th edition Fotomanual (Reg2Class10), or a 4-way classification problem following the 7th edition Fotomanual from 2015 (Class15). AlexNet and the ResNets were pretrained on ImageNet [48] and finetuned on the Sewer-ML subset. Best performance per column is denoted in **bold**. *Adapted from [34].*

Method	**Class10**		**Reg2Class10**		**Class15**	
	micro-F1	**´macro-F1**	**micro-F1**	**macro-F1**	**micro-F1**	**macro-F1**
Random Forest [44]	27.17	23.19	14.63	11.01	68.18	51.47
Extra Trees [45]	29.49	26.39	14.33	10.72	64.34	50.19
AlexNet [46]	30.10	26.96	30.10	28.81	69.59	20.54
ResNet-18 [47]	39.19	**37.41**	**30.61**	**30.00**	73.03	60.93
ResNet-34 [47]	37.37	35.54	28.69	28.00	76.36	61.88
ResNet-50 [47]	**39.70**	36.50	27.07	26.27	**79.29**	**62.88**

$$\text{micro-F1} = 2\frac{\text{micro-Prc} \cdot \text{micro-Rcll}}{\text{micro-Prc} + \text{micro-Rcll}} \quad (4.3) \qquad \text{macro-F1} = \frac{1}{C}\sum_{c=1}^{C} \text{F}_c \quad (4.4)$$

where TP_c, FP_c, FN_c, and F_c are the true positive, false positive, false negative, and the F1-score for class c, and C is the number of annotated classes.

Through this study we found that the performance improved dramatically across all models for the micro-F1 score and macro-F1 scores when following the Class15 setting, indicating a clear benefit in posing the water level classification task as the simpler 4-way classification task, see Table 4.1. Specifically, the two extremes of the scale ([0%-5%) and [30%-100%]) were found to be the easiest to classify, with the intermediate levels being harder to distinguish. We also observed a clear trend of CNNs outperforming the tree-based approaches, even with the small dataset size used, as long as the CNNs were pretrained on ImageNet [48]. Lastly, we saw that training the models in a regression-based fashion did not improve results compared to an 11-way classification-based approach. This was attributed partly to the ordinal nature of the ground truth labels. The study concluded that using data-driven computer vision models is viable for predicting the water level in sewer pipes, as long as the labels are clustered based on visual appearance as per the 7th edition of the Fotomanual.

Building on the knowledge acquired in the previous papers, we set out to investigate the feasibility of not only classifying the sewer defect categories but also the water level and pipe material and shape, *i.e.* all of the classification tasks in the Fotomanual. Specifically, in Paper D we investigated the usefulness of a Multi-Task Classification (MTC) approach, where a single network makes predictions for all four tasks at once. This was the first attempt of using Multi-Task Learning (MTL) in the automated sewer inspection domain as well as classifying the sewer pipe material and shape, which had previously been neglected. Furthermore, we propose a novel MTC approach, the

Cross-Task Graph Neural Network (CT-GNN) Decoder, which leverages the unique fact of the sewer inspection data that all tasks are related, occur concurrently, and a mix of multi-label and multi-class classification tasks.

In order to enable this MTC approach we used the Sewer-ML dataset and augmented with extra labels from the inspection reports. This allowed us to have concurrent labels for the sewer defect, water level, pipe material, pipe shape classification tasks. The labels for the water level classification tasks were converted from the quantity based approach in the 6th edition of the Fotomanual [43] to the visual appearance based approach of the 7th edition of the Fotomanual [49], following the findings of Paper C. Performance on the sewer defect classification task was measured using the $\text{F1}_{\text{Normal}}$ and F2_{CIW} metrics, whereas all other tasks were evaluated using the micro-F1 and macro-F1 metrics. Furthermore, in order to have a single metric for the overall metric performance achieved using the MTC approaches compared to single-task learning (STL) approaches, we used the Δ_{MTL} proposed by Maninis *et al.* [50]. The Δ_{MTL} metric measures the average per-task performance increase for a multi-task model with respect to the STL baselines of the same base architecture:

$$\Delta_{\text{MTL}} = \frac{1}{T}\sum_{t=1}^{T}\frac{(M_{m,t} - M_{b,t})}{M_{b,t}}, \tag{4.5}$$

where $M_{m,t}$ and $M_{b,t}$ are the MLT and STL metric performance for task t, receptively.

MTL and MTC methods often follow an encoder-decoder model structure, where the encoder produces global or per-task feature representation, which the decoder utilize to produce per-task predictions. The CT-GNN decoder is based on the decoder-focused MTL research direction where model parameters are not just shared in the encoder but also in the decoder. This was achieved by producing per-class features for all tasks and refining these features using a Graph Neural Network (GNN) across all classes in all tasks. The underlying weighted directed graph of the Cross-Task GNN was build such that each node represents a class, and the graph weights were determined by either dynamically inferring it at run-time using attention, or an a priori calculated adjacency matrix based on the conditional probability of all pair of classes. This way the network can have a priori known rules encoded directly into its refinement process, such as the deformation (DE) defect only occurring with flexible pipe materials, or intruding roots (RO) and infiltration (IN) often co-occurring with displaced joint (FS). An overview of the CT-GNN approach is shown in Figure 4.1

Two variations of the CT-GNN was tested, CT-GCN and CT-GAT, differing in the GNN used. CT-GCN utilized the seminal Graph Convolutional Network (GCN) [51] building upon an a priori determined adjacency matrix. The adjacency matrix was based on the conditional probability between all classes, with additional processing steps to remove spurious edges and weight the incoming edges. In contrast, CT-GAT utilized the Graph Attention Network (GAT) [52] where the edges are dynamically inferred per example through a self-attention step. However, it was found that limiting the possible edges using a priroi known relations led to a performance increase.

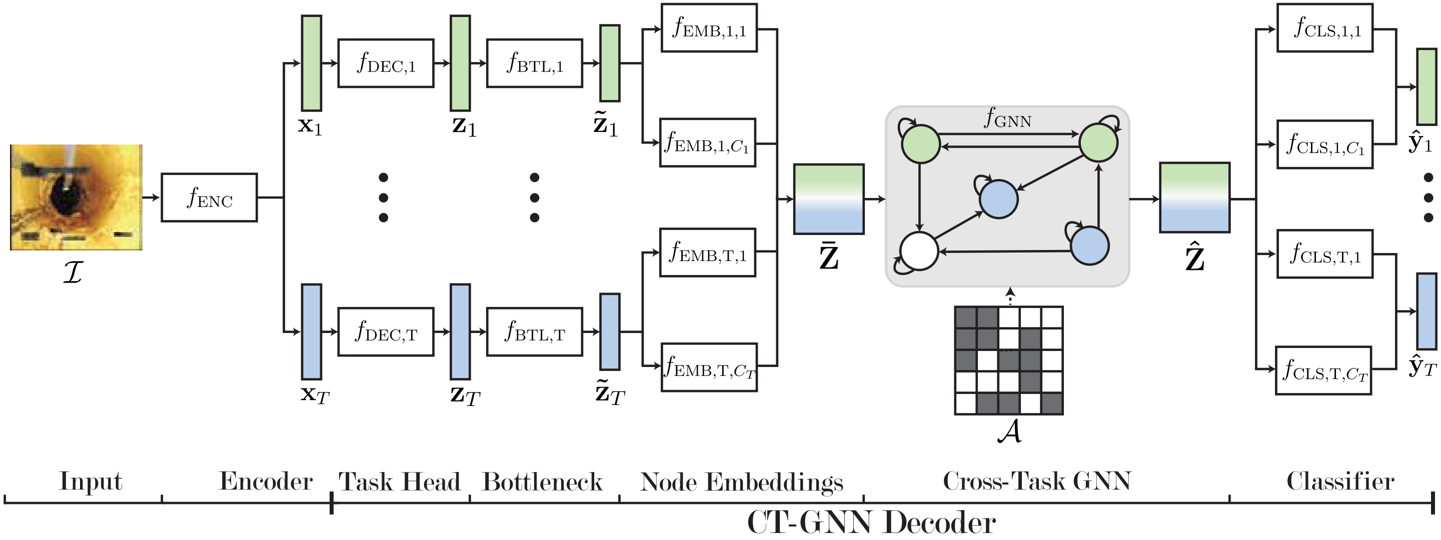


Fig. 4.1: Overview of the proposed CT-GNN MTC approach. Illustration of the Cross-Task Graph Neural Network Decoder for multi-task classification, demonstrating how the CT-GNN decoder head task per-task features and produces per-class features which are subsequently refined using a GNN, where class relationships a determined a priori or dynamically inferred. *Image from [37]*.

Table 4.2: Evaluation of the CT-GNN Decoder, compared to STL and MTL networks. Two version of CT-GNN,using Graph Convolutional Networks (GCN) [51] and Graph Attention Networks (GAT) [52], denoted CT-GCN and CT-GAT respectively, were compared with a hard-shared ResNet-50 encoder (R50-MTL), and the soft-shared MTAN encoder with a ResNet-50 backbone. The performance of the water, shape, and material tasks were measured using the micro-F1 (mF1) and macro-F1 (MF1) metrics. The average per-task improvement with respect to the STL baselines is denoted Δ_{MTL}. #P indicates the number of parameters in millions. * indicates that the method was tested on a subset of the Sewer-ML dataset. Best performance in each column is denoted in **bold**. *Adapted from [37], with the mF1 metrics excluded for brevity.*

	Model		Overall	Defect		Water	Shape	Material
	Model	#P	Δ_{MTL}	**F2_{CIW}**	**$\text{F1}_{\text{Normal}}$**	**MF1**	**MF1**	**MF1**
Validation Split	Benchmark [12]	62.8	-	55.36	91.32	-	-	-
	R50-FT* [34]	23.5	-	-	-	62.53	-	-
	STL	94.0	+0.00	58.42	**92.42**	69.11	46.55	65.99
	R50-MTL	23.5	+10.36	59.73	91.87	70.51	71.64	80.28
	MTAN [53]	48.2	+10.40	61.21	92.10	70.06	68.34	83.48
	CT-GCN	25.2	+12.39	61.35	91.84	**70.57**	**76.17**	82.63
	CT-GAT	24.0	**+12.81**	**61.70**	91.94	**70.57**	74.53	**86.63**
Test Split	Benchmark [12]	62.8	-	55.11	90.94	-	-	-
	R50-FT* [34]	23.5	-	-	-	62.88	-	-
	STL	94.0	+0.00	57.48	**92.16**	69.87	56.15	69.02
	R50-MTL	23.5	+7.39	58.29	91.57	71.17	79.48	**76.35**
	MTAN [53]	48.2	+6.83	59.91	91.72	70.61	78.50	72.73
	CT-GCN	25.2	+7.64	60.07	91.60	70.69	80.32	75.13
	CT-GAT	24.0	**+7.84**	**60.57**	91.61	**71.30**	**81.10**	73.95

In general, the proposed CT-GNN decoder led to better performance than the state-of-the-art performance on the defect and water level classification tasks, as well as outperforming or matching the STL baselines, and the encoder-focused MTC models, see Table 4.2. Specifically, the F2_{CIW}, pipe shape macro-F1, and pipe material macro-F1 scores were improved by 6.34, 29.62, and 21.64 percentage points, respectively. Additionally, the CT-GNN introduces 50 times fewer parameters than the encoder-based Multi-Task Attention Network (MTAN) [53] while achieving a greater performance across all tasks, showing the efficiency of the decoder-based approach.

This work demonstrated the clear benefit of processing and refining the sewer inspection classification tasks simultaneously. Not only did the performance increase across all tasks when using MTC approaches, the CT-GNN also allowed encoding heuristic rules about the relationship between the classes and tasks into the model.

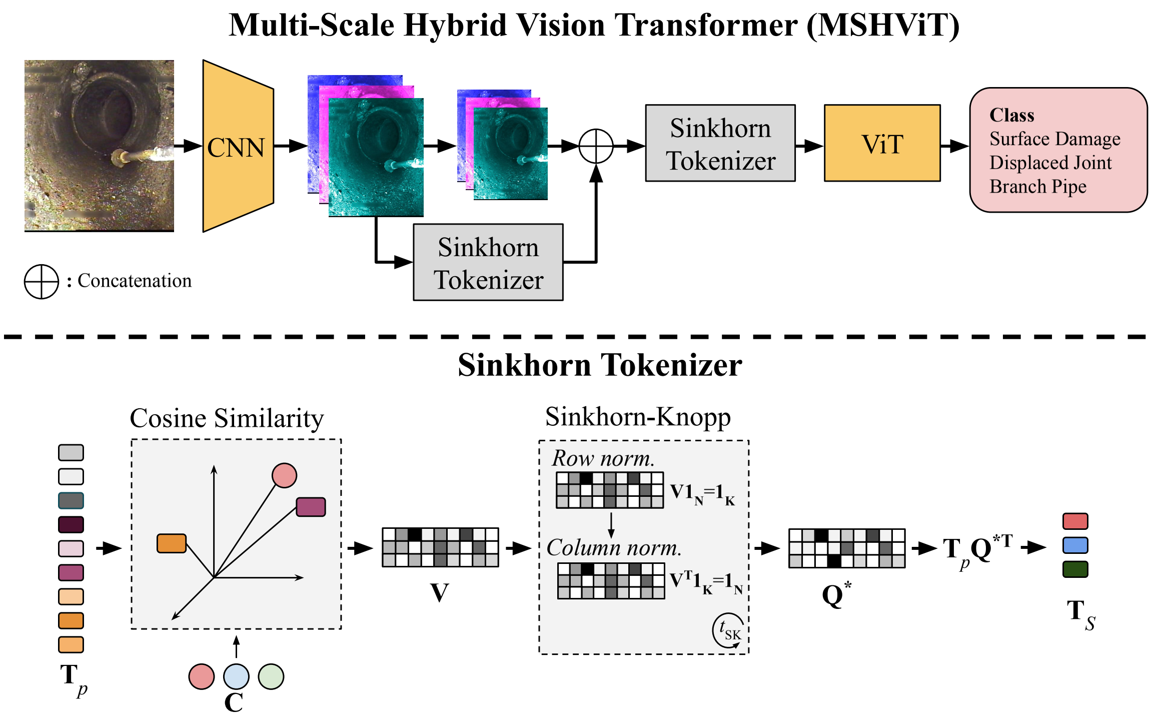
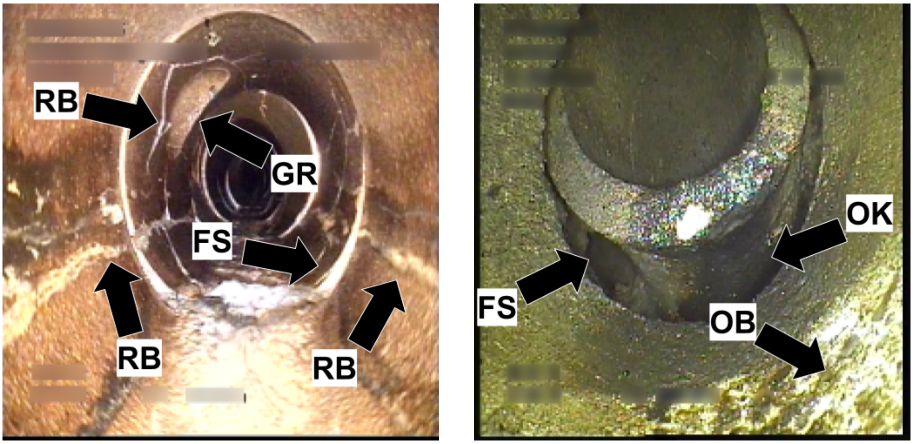


Fig. 4.2: Overview of the proposed MSHViT and Sinkhorn Tokenizer. Illustration of the Multi-Scale Hybrid Vision Transformer and Sinkhorn tokenizer for sewer defect classification, showing how the features from different scales are extracted, cluster using the Sinkhorn tokenizer, and propagated forward. *Image from [20].*



(a) Defects: **cracks, breaks, and collapses (RB)**, **displaced joint (FS)**, and **branch pipe (GR)**.

(b) Defects: **surface damage (OB)**, **displaced joint (FS)**, and **connection with construction changes (OK)**.

Fig. 4.3: Example of how defects can be at multiple places and span over long ranges in image space, leading to non-local interactions across multiple scales. *Images from the Sewer-ML dataset [12].*

Lastly, we shifted our focus back to the defect category classification task. Paper E builds upon the observation that non-local spatial semantics are critical for capturing relationships between scales and defects in a classification setting. As a motivating example, see Figure 4.3 where two sewer images are shown. In Figure 4.3a there are cracks along the pipe wall and a displaced joint, both stretching over a large part of the image space at multiple places. Similarly, in Figure 4.3b the surface damage and displaced joint are not constrained to a single local area of the image, instead showing long range dependencies.

We incorporated this information into our models by building upon the recently proposed Hybrid Vision Transformer (HViT) [21]. Instead of applying a Transformer directly onto the image, HViT applies the Transformer onto the last feature map from a CNN, effectively forming a fully-connected graph with dynamic weights between the tokens (*i.e.* feature vectors). This way the strong spatial inductive bias of the CNN is leveraged while allowing for non-local interaction through the Transformer architecture. First, we proposed the Multi-Scale Hybrid Vision Transformer (MSHVIT), which considers multi-scale information such as cracks running along the pipe wall, by aggregating features from different stages of a CNN, and propagating tokens progressively across scales. Secondly, we hypothesized that tokens from the same region encode similar information, leading to redundant token representation. We circumvented this by introducing the Sinkhorn Tokenizer, a clustering-based tokenization method where the feature vectors are clustered using the Sinkhorn-Knopp algorithm [54]. The full proposed architecture is illustrated in Figure 4.2.

Using the Sewer-ML defect classification dataset we demonstrated that the MSHVIT and Sinkhorn tokenizer achieves a significant performance improvement when com-

Table 4.3: Comparison of MSHViT to baseline models and HViT-like models. Comparison of using MSHViT with different backbones, as well as difference in performance to previous published results on Sewer-ML [12, 37], and HViT-like models [21, 55, 56]. Best performance per column is denoted in **bold**. *Adapted from [20].*

Model		Validation Split		Test Split	
Model	**MSHViT**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**
Benchmark [12]	-	55.36	91.32	55.11	90.94
CT-GAT [37]	-	**61.70**	91.94	**60.57**	91.61
ResNet-50-HViT-Patch [21]	-	59.87	92.41	57.58	91.99
ResNet-50-HViT-Sinkhorn [21]	-	60.42	92.41	58.74	92.07
BotNet-50-S1 [55]	-	61.62	**92.92**	59.69	**92.49**
CoAtNet-0 [56]	-	57.82	92.28	56.53	91.94
CoAtNet-1 [56]	-	59.37	92.50	57.42	91.11
ResNet-18 [47]	✗	58.60	92.34	56.62	91.88
	✓	59.87	92.42	58.18	92.12
ResNet-34 [47]	✗	60.98	92.72	59.18	92.30
	✓	61.65	92.76	59.91	92.30
ResNet-50 [47]	✗	59.28	92.44	57.58	92.03
	✓	61.68	92.44	60.11	92.11
ResNet-101 [47]	✗	60.06	92.48	58.01	92.13
	✓	61.25	92.50	59.93	92.19
TResNet-M [57]	✗	58.04	92.22	56.08	91.90
	✓	58.68	92.25	56.93	91.84
TResNet-L [57]	✗	59.17	92.36	56.97	92.00
	✓	59.19	92.27	57.16	91.87

pared to the CNN baseline as well as other HViT inspired models, see Table 4.3. We observe clear improvements on the ResNet and TResNet backbones [47, 57], while outperforming the conventional HViT model, as well as newer HViT-like models such as BoTNet-50-S1 [55] and CoAtNets [56]. Using the ResNet-50 backbone we saw that the F2$_{\text{CIW}}$ score improved by 2.53 percentage points under equal training conditions, and matches the performance of the CT-GAT without using information from other concurrent classification tasks. We qualitatively verified that the Sinkhorn tokenizer was capable of capturing non-local spatial semantics, and found that the Sinkhorn tokenizer resulted in an increased efficiency compared to the conventional non-overlapping patch tokenizer when measuring training and inference throughput.

3 Contributions

We have furthered the image-based automation of sewer inspections field by improving performance on not only the sewer defect classification task, but also on the previously ignored water level, pipe material, and pipe shape tasks. This progress has been achieved by using the multi-task classification framework within the sewer inspection field for the first time, developing a novel multi-task classification decoder head where domain knowledge and heuristic rules can be directly incorporated to refine features across tasks, and extending the hybrid vision transformer with multi-scale information and a novel clustering-based tokenizer which achieving competitive results on Sewer-ML with a $F2_{\text{CIW}}$ score of 60.11%. Our main contributions within the advancement of the image-based automation of sewer inspections field are thus:

- An empirical investigation to determine how to encode water level information, showing the effectiveness of appearance based labels, rather than exact quantities, as used in the 7th edition Fotomanual.
- The first use of the multi-class classification framework for sewer classification tasks, demonstrating significant improvements over single task networks on the sewer pipe shape and material classification tasks.
- The novel Cross-Task Graph Neural Network (CT-GNN) decoder for multi-task classification, enabling efficient information sharing and incorporation of heuristic rules and domain knowledge.
- The novel Multi-Scale Hybrid Vision Transformer (MSHViT) architecture for sewer defect classification, extending the hybrid vision transformer with cross-scale information and a clustering-based Sinkhorn tokenizer.
- State-of-the-Art sewer defect classification with a $F2_{\text{CIW}}$ score of 60.57% using CT-GNN and competitive $F1_{\text{Normal}}$ score of 92.30% using MSHViT.
- State-of-the-Art water level classification with a macro-F1 score of 71.30%.
- State-of-the-Art pipe shape classification with a macro-F1 score of 81.10%.
- A competitive performance on the pipe material classification task, with a macro-F1 score of 73.95%.

References

[1] J. B. Haurum and T. B. Moeslund, “A survey on image-based automation of cctv and sset sewer inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[2] J. C. Cheng and M. Wang, “Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,” *Automation in Construction*, vol. 95, pp. 155–171, 2018.

[3] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, “Deep learning based automated detection of sewer defects in cctv videos,” *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[4] M. Wang, S. S. Kumar, and J. C. Cheng, “Automated sewer pipe defect tracking in cctv videos based on defect detection and metric learning,” *Automation in Construction*, vol. 121, p. 103438, 2021.

[5] M. Wang, H. Luo, and J. C. Cheng, “Towards an automated condition assessment framework of underground sewer pipes based on closed-circuit television (cctv) images,” *Tunnelling and Underground Space Technology*, vol. 110, p. 103840, 2021.

[6] D. Li, Q. Xie, Z. Yu, Q. Wu, J. Zhou, and J. Wang, “Sewer pipe defect detection via deep learning with local and global feature fusion,” *Automation in Construction*, vol. 129, p. 103823, 2021.

[7] Q. Zhou, Z. Situ, S. Teng, W. Chen, G. Chen, and J. Su, “Comparison of classic object-detection techniques for automated sewer defect detection,” *Journal of Hydroinformatics*, 03 2022.

[8] Q. Zhou, Z. Situ, S. Teng, and G. Chen, “Convolutional neural networks&#x2013;based model for automated sewer defects detection and classification,” *Journal of Water Resources Planning and Management*, vol. 147, no. 7, p. 04021036, 2021.

[9] C. Siu, M. Wang, and J. C. Cheng, “A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection,” *Automation in Construction*, vol. 137, p. 104213, 2022.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.

[11] L. M. Dang, H. Wang, Y. Li, T. N. Nguyen, and H. Moon, “Defecttr: End-to-end defect detection for sewage networks using a transformer,” *Construction and Building Materials*, vol. 325, p. 126584, 2022.

[12] J. B. Haurum and T. B. Moeslund, “Sewer-ml: A multi-label sewer defect classification dataset and benchmark,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[13] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua, “Gated context aggregation network for image dehazing and deraining,” *WACV 2019*, 2018.

[14] M. Abdullah-Al-Wadud, M. H. Kabir, M. A. Akber Dewan, and O. Chae, “A dynamic histogram equalization for image contrast enhancement,” *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 593–600, 2007.

[15] G. Pan, Y. Zheng, S. Guo, and Y. Lv, “Automatic sewer pipe defect semantic segmentation based on improved u-net,” *Automation in Construction*, vol. 119, p. 103383, 2020.

[16] M. Wang and J. C. P. Cheng, “A unified convolutional neural network integrated with conditional random field for pipe defect segmentation,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, pp. 162–177, 2020.

[17] Y. Li, H. Wang, L. Dang, M. Jalil Piran, and H. Moon, “A robust instance segmentation framework for underground sewer defect detection,” *Measurement*, vol. 190, p. 110727, 2022.

[18] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, “SOLOv2: Dynamic and fast instance segmentation,” in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[19] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[20] J. B. Haurum, M. Madadi, S. Escalera, and T. B. Moeslund, “Mshvit: Multi-scale hybrid vision transformer and sinkhorn tokenizer for sewer defect classification,” *Machine Vision and Applications*, 2022, Under Review.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.

[22] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114.

[23] Y. Gu, W. Tu, Q. Li, T. Zhao, D. Zhao, S. Zhu, and J. Zhu, “Collaboratively inspect large-area sewer pipe networks using pipe robotic capsules,” in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 211–220.

[24] M. Klusek and T. Szydlo, “Supporting the process of sewer pipes inspection using machine learning on embedded devices,” in *Computational Science – ICCS 2021*, M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham: Springer International Publishing, 2021, pp. 347–360.

[25] S. Yang, Z. Zhao, Q. Yang, and J. Wang, “Attention guided image enhancement network for sewer pipes defect detection,” in *2021 4th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, 2021, pp. 109–113.

[26] L. M. Dang, S. Kyeong, Y. Li, H. Wang, T. N. Nguyen, and H. Moon, “Deep learning-based sewer defect classification for highly imbalanced dataset,” *Computers & Industrial Engineering*, vol. 161, p. 107630, 2021.

[27] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[29] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, “Analyzing and improving the image quality of stylegan,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[30] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, "Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known cnn classifier," *IEEE Access*, vol. 9, pp. 59 498–59 507, 2021.

[31] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, and J. Dong, "A multi-defect detection system for sewer pipelines based on stylegan-sdm and fusion cnn," *Construction and Building Materials*, vol. 312, p. 125385, 2021.

[32] H. W. Ji, S. S. Yoo, B.-J. Lee, D. D. Koo, and J.-H. Kang, "Measurement of wastewater discharge in sewer pipes using image analysis," *Water*, vol. 12, no. 6, 2020.

[33] H. W. Ji, S. S. Yoo, D. D. Koo, and J.-H. Kang, "Determination of internal elevation fluctuation from cctv footage of sanitary sewers using deep learning," *Water*, vol. 13, no. 4, 2021.

[34] J. B. Haurum, C. H. Bahnsen, M. Pedersen, and T. B. Moeslund, "Water level estimation in sewer pipes using deep convolutional neural networks," *Water*, vol. 12, no. 12, 2020.

[35] F. Plana Rius, M. P. Philipsen, J. M. Mirats Tur, T. B. Moeslund, C. Angulo Bahón, and M. Casas, "Autoencoders for semi-supervised water level modeling in sewer pipes with sparse labeled data," *Water*, vol. 14, no. 3, 2022.

[36] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[37] J. B. Haurum, M. Madadi, S. Escalera, and T. B. Moeslund, "Multi-task classification of sewer pipe defects and properties using a cross-task graph neural network decoder," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 2806–2817.

[38] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004, british Machine Vision Computing 2002.

[39] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, "Underground sewer pipe condition assessment based on convolutional neural networks," *Automation in Construction*, vol. 106, p. 102849, 2019.

[40] L. M. Dang, S. I. Hassan, S. Im, I. Mehmood, and H. Moon, "Utilizing text recognition for the defects extraction in sewers cctv inspection videos," *Computers in Industry*, vol. 99, pp. 96–109, 2018.

[41] S. Moradi, T. Zayed, F. Nasiri, and F. Golkhoo, "Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning& based text recognition," *Journal of Infrastructure Systems*, vol. 26, no. 3, p. 04020018, 2020.

[42] Q. Zhou, Z. Situ, S. Teng, H. Liu, W. Chen, and G. Chen, "Automatic sewer defect detection and severity quantification based on pixel-level semantic segmentation," *Tunnelling and Underground Space Technology*, vol. 123, p. 104403, 2022.

[43] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 6th ed. Dansk Vand og Spildevandsforening (DANVA), 2010.

[44] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[45] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 3 2006.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[47] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, pp. 770–778.

[48] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[49] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 7th ed. Dansk Vand og Spildevandsforening (DANVA), 2015.

[50] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, “Attentive single-tasking of multiple tasks,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[52] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph Attention Networks,” *International Conference on Learning Representations*, 2018.

[53] S. Liu, E. Johns, and A. J. Davison, “End-to-end multi-task learning with attention,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1871–1880.

[54] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[55] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, “Bottleneck transformers for visual recognition,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 514–16 524.

[56] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “Coatnet: Marrying convolution and attention for all data sizes,” *CoRR*, vol. abs/2106.04803, 2021. [Online]. Available: https://arxiv.org/abs/2106.04803

[57] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, “Tresnet: High performance gpu-dedicated architecture,” *CoRR*, 2020.

Chapter 5

Point Cloud-based Automation of Sewer Inspections

The CCTV sensor has been the primary sensor for sewer inspections throughout the years, as it is easily interpretable for humans, and cheap due to mass production. However, this is not equivalent to it being the best sensor for automated sewer inspections. Throughout the years, several alternative sensors have been investigated, such as thermal imaging to detect infiltration through the pipe walls, ultrasound through the pipe material to determine structural defects, or ground penetrating radar to determine the state of the soil surrounding the pipes [1, 2]. All of these alternative modalities provide information that is otherwise not available when simply using a CCTV sensor. However, the sensors are rarely easily interpretable for human inspectors, limiting their use in current sewer inspections.

Alternative sensing methodologies have, however, been included in nearly all modern robotic sewer inspection solutions [3–9]. Specifically, the depth modality is often utilized as it provides valuable information about the structure of the sewer pipe, which is not immediately obvious using a CCTV sensor. Furthermore, the depth sensor is more versatile and less vulnerable to environmental factors than thermal or acoustic based sensors, while being more easy to interpret. Specifically, the depth data can typically be presented as a 2D depth map, or a 3D point cloud. If the 3D data is further fused with RGB information, the sewer inspectors can comfortably navigate the extra modality information. Lastly, the depth data is not only usable for defect detection, but can also be integrated into the navigation and localization algorithms of the robot as well as used to generate reconstructions of the pipes, which can be valuable for off-site analysis.

In the following, we give a short introduction into how deep learning has been used to work with depth data, as well as how depth data has been used within the automated sewer inspection domain.

1 State-of-the-Art

Wile other methods have been applied on 3D data in the sewer context, these methods have been developed in the context of the chosen sensors. As deep learning-based methods have recently been used, we focus on reviewing the deep learning-based approaches from the general computer vision field.

The first investigations into using deep learning with point cloud data were divided into two main branches: Volumetric CNN where 3D convolutions were used in CNNs, requiring the continuous space to be discretized into a voxel based structure [10–16], and multi-view CNNs, where several views of the point cloud or model mesh were provided and a shared CNN produces a feature vector per view, which were lastly pooled and processed with an MLP [15, 17, 18]. These methods are, however, hampered by intrinsic limitations. Volumetric based approaches tend to lead to overly dense and sparse areas, as the scenes are voxelized without considering the density of different regions. Multi-view CNNs are in contrast limited by the need for processing several renderings of the object in order to make a prediction, as occlusions become prevalent when forcing the projecting of the 3D point cloud or mesh into a 2D image. Furthermore, the viewpoints are typically predetermined even if this does not lead to the best performance. However, recent work has proposed a differentiable rendering approach where a neural network determines the camera attributes based on the input point cloud [18].

Meanwhile, the research branch of Geometric Deep Learning (GDL) has emerged, focused on studying the effect of symmetries in neural networks and how to generalize to non-Euclidean domains such as graphs, manifolds, sets, *etc*. [19, 20]. Specifically, GDL can be said to be the study of incorporating the relevant geometric priors of the data structure into the network using appropriate equivariant and invariant functions, and generalizing concepts such as convolutions to non-Euclidean domains [21–24]. For sets and graphs this could be the permutation invariance of the elements, while for data structures such as images and point clouds (*i.e*. an unordered set) it could be the equivariance to the rotation groups SO(2) and SO(3), respectively [25, 26]. The first NN to directly process a point cloud was PointNet [27], where a shared MLP processes each point separately (with only the 3D position and optionally normal vector as input) and enforced permutation invariance using global pooling layers. However, by construction PointNet only considered local information, as each point is processed independently. Subsequent work such as PointNet++ [28] included global information by a hierarchical set abstraction where the points are sequentially sampled, grouped, and processed, or as in the Dynamic Graph CNN (DGCNN) [29] where for each point a local directed graph consisting of k nodes is constructed using k-Nearest Neighbors (kNN) in feature space and node features aggregated with a symmetric function. This has spawned a large amount of work [25, 30–33], with the most recent methods utilizing the Transformer architecture to incorporate global information through a self-attention based dynamic graph [34–37].

As previously mentioned, depth sensors have commonly been used within the sewer inspection domain in order to obtain extra information which is either impossible or hard to obtain with a CCTV sensor. The depth sensor has been commonly used in the sensor suite of many researched robotic platforms for sewer inspections, from stereo IR sensors and laser scanner on the KANTARO robot [7], to RGBD sensors on the ARSI and SIAR robots [5, 8], and most recently the Time-of-Flight (ToF) as well as active and passive stereo cameras in the ASIR research project [6]. The primary use of these sensors have for a long time been navigation and control of the robot [5, 8, 38, 39] as well as reconstruction of the pipe [6, 8, 40]. In contrast, the use of the depth data for classifying defect categories has for the longest time been relegated to a niche interest.

Duran *et al.* [41–43] presented one of the first uses of laser scanners detecting defects, where a laser diode project a ring of light onto the sewer pipe wall in an non-illuminated pipe, and registered with a CCD sensor to obtain an intensity map. When combined with positional data from a robot the intensity map can provide 3D data of the pipe geometry. Using an image processing based pipeline the intensity values along the projected ring are extracted. Using an MLP these intensity signatures were shown to be useful for both binary and multi-class defect classification, with different severity levels included. Tezerjani *et al.* [44] similarly use these intensity maps, but instead fits a B-spline to the extracted boundaries and designating outliers from this spline to be defects. In parallel, Lepot *et al.* [45] constructed a thorough overview of the feasibility of using laser scanners to extract the different defects in the European inspection standard in comparison to image-based defect classification. Through this study they found that the laser scanner can be used to quantify a majority of the defects, and accurately determine the diameter, shape, and roughness of the pipe. Alejo *et al.* [46, 47] used the RGBD sensor in the SIAR robot to help with localization in the poorly illuminated and GPS-denied sewer pipes, by classifying manholes using a lightweight CNN trained on depth images. Merino *et al.* [48] further developed a segmentation network for the SIAR robot, where the RGBD data was converted to a point cloud, and segmented by aligning the recorded 3D data with a bank of 3D models of known types of pipe sections using the Iterative Closest Point algorithm [49]. As the 3D models are annotated beforehand, this allows for easy segmentation of the pipe walls into distinct structural features such as left/right wall, roof, gutter *etc*. This approach also allows for detecting structural defects through the alignment error, as a high alignment error indicates significant difference between the observed sewer and the known pipe section types. A major downside of the approaches applied to laser scanning and RGBD data is that except for the segmentation method by Merino *et al.* [48], all other applied approaches do not utilize the geometry of the observed scene which is encoded in the 3D data, instead nearly always opting to use 2D projections of the data.

However, Haurum *et al.* [50] who worked directly on point cloud data, and compared the PointNet and DGCNN methods on a four-way multi-class classification dataset, which was made publicly available. This dataset was constructed by combining a small amount of manually recorded point clouds from a laboratory setting, together with several thousand point clouds from a modified synthetic sewer point cloud generator first proposed by Henriksen *et al.* [51]. Through this data it was demonstrated that pre-training on the synthetic data followed by fine-tuning on the real data leads to a decent results (an F1-score of 23.58%), similar to the observations made in the image-based sewer inspection field [52]. The work was further built upon by Zhou *et al.* [37], who proposed a Transformer based architecture for point cloud classification, denoted TransPCNet, based on a kNN feature embedding similar to DGCNN and stacked self-attention layers. Combined with a weighted and label-smoothed cross-entropy loss, TransPCNet dramatically outperforms both PointNet and DGCNN by achieving an F1-score of 60.58%. This not only shows the power of the TransPCNet, but also the need for future dataset development within the point cloud-based sewer inspection field. The currently used dataset, while a good stepping stone, is recorded in idealistic conditions with heavily reduced complexity, as all pipes are dry plastic pipes with one defect at a time. However, the largest hurdle within this area of sewer inspection automation is the collection of data. Unlike the image-based automation field, there are rarely any depth sensors mounted on the sewer inspection tracktors and therefore a lack of historic annotated data.

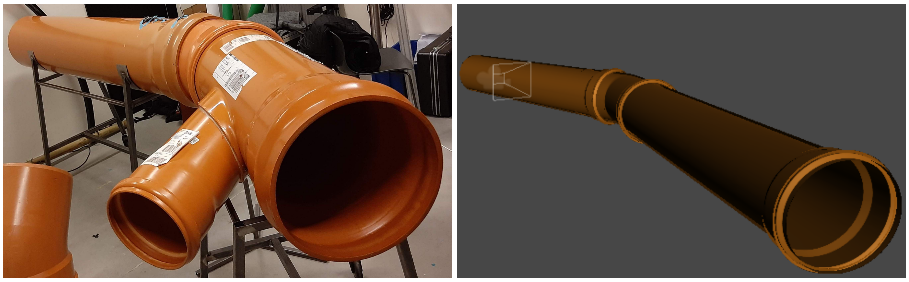
2 Contributed Scientific Work

As a part of the ASIR research project investigation on using the depth sensing modality, this Ph.D. thesis has focused on how to utilize synthetic point cloud data in order to circumvent the lack of data in the point cloud-based automation of sewer inspections field. This has resulted in two papers published in computer vision conferences, see appendix F – G.

In paper F, we conduct an initial investigation into how a synthetic sewer pipe data can be constructed in a systematic manner.

An open-source framework was built in Unity based on the principle of Structured Domain Randomization (SDR) [53], originally proposed for constructing plausible synthetic data from a vehicular view point. This allowed placing predefined objects (*i.e.* pipes and defects) along a randomly generated spline, by sampling the objects according to a probability distribution conditioned on the global and local context. For example, the intruding sealing material (IS) defect can only plausibly occur at joints, which should be reflected by the simulation. The Camboard Pico Flexx [54] Time-of-Flight (ToF) sensor chosen for the ASIR project was simulated in order to ensure compatibility between the real and synthetic data. This was achieved using an approximate simulation using ray-tracing, with camera parameters and precision uncertainty based on the available sensor datasheet. Similarly, the framework was restricted to dry clean plastic pipes in order to simplify the interaction between camera rays and the pipe material and environment.

The constructed framework was validated by constructing two controlled pipe



(a) Physical setup.

(b) Virtual setup.

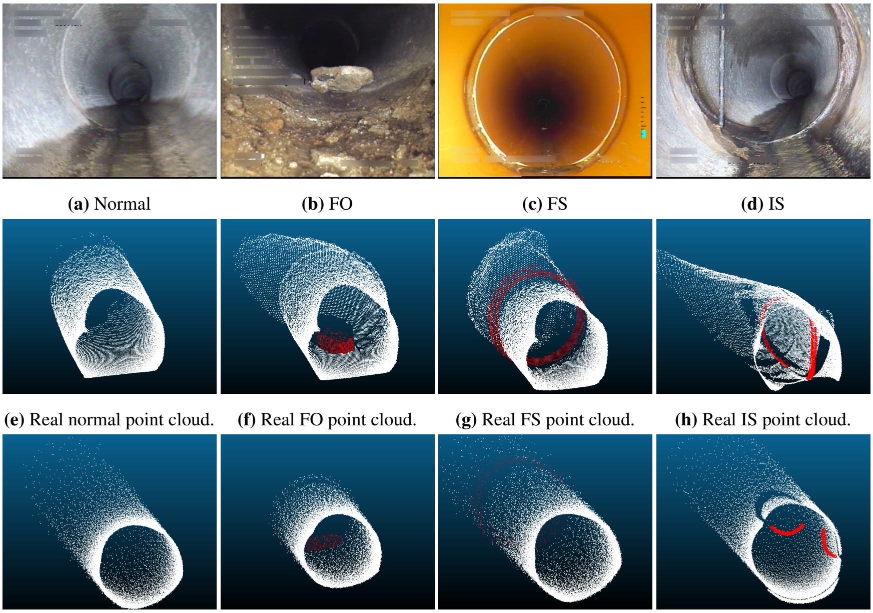
Fig. 5.1: Images of the laboratory test site and the virtual twin, used to determine the quality of the simulated ToF sensor. Note that the branch pipe was not replicated in the virtual setup, as the camera was moved through the long straight pipe seen further away, and recording stopped before the branch pipe was in view. *Reproduced from [51].*

setups in a laboratory with and without a displaced joint, and manually modeling the setups in Unity. One of the pipe setups are shown in Figure 5.1. Data was collected by moving the sensor (virtual and physical) through the pipes, capturing point clouds at different time steps. It was evaluated that the absolute difference between the synthetic and real point clouds were 5.78 ± 8.92 mm and 7.58 ± 8.68 mm for each of the two test scenarios. As the pipe diameter was 376 mm, this indicates that the simulated data is within a tolerable margin of error. Similarly, it was found that the measured diameter of the synthetic point clouds tended to be closer to the real pipe diameter, whereas the diameter of the real point clouds overestimate the diameter of the pipe. This is hypothesized to be due to factors such as surface imperfections, roughness, and reflections interfering with the ToF sensor. This work demonstrated the feasibility of simulating synthetic point cloud data under reasonable constraints in scene variability.

In Paper G, we investigated usefulness of the synthetic data generator for bootstrapping the training process of models for point cloud-based automation of sewer inspections.

We compared two commonly used geometric deep learning models, PointNet [27] and DGCNN [29], trained under four different data scenarios: (1) only using synthetic data, (2) only using real data, (3) training on synthetic and real data, and (4) training on synthetic data and fine-tune on real data. A hyperparameter grid search was employed for each data scenario in order to determine the best performing learning rate and weight decay for each model.

To facilitate these tests a multi-class classification dataset with 17,027 point clouds and four classes (normal non-defective pipes, and pipes with either displaced joint (FS), intruding sealing materials (IS), or obstacles/bricks (FO)) were created. The previously developed synthetic data generator was expanded upon in order to generate these classes, and similarly a real life dataset was collected in a laboratory environment. An example of the used point clouds can be observed in Figure 5.2 with visual example


(a) Normal **(b)** FO **(c)** FS **(d)** IS

(e) Real normal point cloud. **(f)** Real FO point cloud. **(g)** Real FS point cloud. **(h)** Real IS point cloud.

(i) Synth. normal point cloud. **(j)** Synth. FO point cloud. **(k)** Synth. FS point cloud. **(l)** Synth. IS point cloud.

Fig. 5.2: Examples of pipes with no defects (normal), an obstacle (FO), a displaced joint (FS), and intruding sealing material (IS) from actual sewer inspections, as well as point clouds for each class either recorded in a laboratory in dry plastic pipes or synthetically. Defects in the point clouds are annotated in red. *Images from [50] and the Sewer-ML dataset [55].*

of each considered class from real sewer inspections. The dataset was constructed such that the 485 of the 827 real point clouds were placed in the test split, in order to investigate the effectiveness of predominantly using synthetic training data.

Through our experiments we found that the DGCNN network regularly outperformed the PointNet network, across all data scenarios, see Table 5.1. This is not immediately obvious from the F1-scores, however, when investigating the per-class performance we see that the PointNet models always ignores one or more classes in its predictions. Similar behavior is found for the DGCNN-S2 model. Similarly, we found that the synthetic data was best utilized when the models were initially trained only on the synthetic data and then fine-tuned on real data. However, while the DGCNN performs best in our tests it still only achieves a class-weighted F1-score of 23.58% on the real point clouds in the test split, indicating a clear need for further development within this field. To facilitate this research direction the dataset was made publicly available.

Table 5.1: Benchmarking point-cloud based sewer defect classification networks. Performance of the PointNet and DGCNN networks on the real data test split, and the combined test split, for all four data scenarios. All metrics are the weighted average across all classes. *Adapted from [50].*

Model-Scenario	Real Data			Synthetic & Real Data		
	Precision	Recall	F1	Precision	Recall	F1
PointNet-S1	3.58	15.88	5.25	8.00	17.21	6.70
DGCNN-S1	29.02	20.62	17.65	57.57	56.73	57.09
PointNet-S2	2.72	16.49	4.67	2.77	16.64	4.75
DGCNN-S2	25.31	50.31	33.68	25.05	50.05	33.39
PointNet-S3	28.61	32.16	30.23	34.36	32.40	31.65
DGCNN-S3	34.55	22.27	16.66	58.72	57.52	58.67
PointNet-S4	23.17	27.42	24.24	28.37	36.11	30.98
DGCNN-S4	39.69	26.19	23.58	50.37	36.61	35.25

3 Contributions

We have investigated the usefulness of synthetic data for point cloud-based sewer defect classification, as point cloud data from real sewer pipes is scarce. Using a synthetic point cloud generator and small amount of real point cloud data from a laboratory setup, we found that synthetic point cloud data is a viable option for bootstrapping the training procedure as long a some real data is available for fine-tuning. Our main contributions within the field of point cloud-based automation of sewer inspections are thus:

- The development of an open-source synthetic data generator based on the Structured Domain Randomization principle, allowing for easy sampling of varied synthetic sewer point clouds.
- The world's first publicly available point cloud-based sewer inspection dataset framed as a four-way multi-class classification task, containing a total of 17,027 real and synthetic sewer point clouds.
- The comparison of two commonly used geometric deep learning models, PointNet and DGCNN, trained under a set of different data scenarios. The DGCNN model trained on synthetic data and fine-tuned on real data achieved an F1-score of 23.58% on real sewer point clouds. This serves as a baseline for future research in the field.

References

[1] Z. Liu and Y. Kleiner, “State of the art review of inspection technologies for condition assessment of water pipes,” *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

[2] O. Duran, K. Althoefer, and L. D. Seneviratne, “State of the art in sensor technologies for sewer inspection,” *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, April 2002.

[3] M. Kolesnik and H. Streich, “Visual orientation and motion control of makro-adaptation to the sewer environment,” in *In Proceedings of the Seventh International Conference on the Simulation of Adaptive Behavior*, vol. 4, 2002, pp. 62–69.

[4] F. Kirchner and J. Hertzberg, “A prototype study of an autonomous robot platform for sewerage system maintenance,” *Autonomous Robots*, vol. 4, pp. 319–331, 1997.

[5] D. Alejo, G. Mier, C. Marques, F. Caballero, L. Merino, and P. Alvito, *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*. Cham: Springer International Publishing, 2020, pp. 275–296.

[6] C. H. Bahnsen, A. S. Johansen, M. P. Philipsen, J. W. Henriksen, K. Nasrollahi, and T. B. Moeslund, “3d sensors for sewer inspection: A quantitative review and analysis,” *Sensors*, vol. 21, no. 7, 2021.

[7] A. A. F. Nassiraei, Y. Kawamura, A. Ahrary, Y. Mikuriya, and K. Ishii, “Concept and design of a fully autonomous sewer pipe inspection mobile robot "kantaro",” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 136–143.

[8] F. Chataigner, P. Cavestany, M. Soler, C. Rizzo, J.-P. Gonzalez, C. Bosch, J. Gibert, A. Torrente, R. Gomez, and D. Serrano, *ARSI: An Aerial Robot for Sewer Inspection*. Cham: Springer International Publishing, 2020, pp. 249–274.

[9] J. M. Mirats Tur and W. Garthwaite, “Robotic devices for water main in-pipe inspection: A survey,” *Journal of Field Robotics*, vol. 27, no. 4, pp. 491–508, 2010.

[10] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, “Generative and discriminative voxel modeling with convolutional neural networks,” *CoRR*, vol. abs/1608.04236, 2016.

[11] D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[12] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[13] Y. Zhou and O. Tuzel, “Voxelnet: End-to-end learning for point cloud based 3d object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum pointnets for 3d object detection from rgb-d data,” *arXiv preprint arXiv:1711.08488*, 2017.

[15] C. R. Qi, H. Su, M. Niessner, A. Dai, M. Yan, and L. J. Guibas, “Volumetric and multi-view cnns for object classification on 3d data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] Y. Yan, Y. Mao, and B. Li, “Second: Sparsely embedded convolutional detection,” *Sensors*, vol. 18, no. 10, 2018.

[17] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, “Multi-view convolutional neural networks for 3d shape recognition,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[18] A. Hamdi, S. Giancola, and B. Ghanem, “Mvtn: Multi-view transformation network for 3d shape recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 1–11.

[19] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, “Geometric deep learning: Going beyond euclidean data,” *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[20] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic, “Geometric deep learning: Grids, groups, graphs, geodesics, and gauges,” *CoRR*, vol. abs/2104.13478, 2021.

[21] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[22] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[23] T. S. Cohen, M. Geiger, J. Köhler, and M. Welling, “Spherical CNNs,” in *International Conference on Learning Representations*, 2018.

[24] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling, “Gauge equivariant convolutional networks and the icosahedral CNN,” in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 1321–1330.

[25] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas, “Vector neurons: A general framework for so(3)-equivariant networks,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 200–12 209.

[26] M. W. Lafarge, E. J. Bekkers, J. P. Pluim, R. Duits, and M. Veta, “Roto-translation equivariant convolutional networks: Application to histopathology image analysis,” *Medical Image Analysis*, vol. 68, p. 101849, 2021.

[27] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[29] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic graph cnn for learning on point clouds,” *ACM Trans. Graph.*, vol. 38, no. 5, Oct. 2019.

[30] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual MLP framework," in *International Conference on Learning Representations*, 2022.

[31] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[32] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[33] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[34] C. Zhang, H. Wan, S. Liu, X. Shen, and Z. Wu, "Pvt: Point-voxel transformer for 3d deep learning," *arXiv preprint arXiv:2108.06076*, 2021.

[35] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268.

[36] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, pp. 187–199, Apr. 2021.

[37] Y. Zhou, A. Ji, and L. Zhang, "Sewer defect detection from 3d point clouds using a transformer-based deep learning model," *Automation in Construction*, vol. 136, p. 104163, 2022.

[38] A. Ahrary, Y. Kawamura, and M. Ishikawa, "A laser scanner for landmark detection with the sewer inspection robot kantaro," in *2006 IEEE/SMC International Conference on System of Systems Engineering*, 2006, pp. 6 pp.–.

[39] D. Alejo, F. Chataigner, D. Serrano, L. Merino, and F. Caballero, "Into the dirt: Datasets of sewer networks with aerial and ground platforms," *Journal of Field Robotics*, vol. 38, no. 1, pp. 105–120, 2021.

[40] K. Kawasue and T. Komatsu, "Shape measurement of a sewer pipe using a mobile robot with computer vision," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, p. 52, 2013.

[41] O. Duran, K. Althoefer, and L. Seneviratne, "Pipe inspection using a laser-based transducer and automated analysis techniques," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 3, pp. 401–409, 2003.

[42] ——, "Automated pipe inspection using ann and laser data fusion," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, 2004, pp. 4875–4880 Vol.5.

[43] O. Duran, K. Althoefer, and L. D. Seneviratne, "Automated pipe defect detection and categorization using camera/laser-based profiler and artificial neural network," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 118–126, 2007.

[44] Tezerjani, Abbasali Dehghan, Mehrandezh, Mehran, and Paranjape, Raman, “Defect detection in pipes using a mobile laser-optics technology and digital geometry,” *MATEC Web of Conferences*, vol. 32, p. 06006, 2015.

[45] M. Lepot, N. Stanić, and F. H. Clemens, “A technology for sewer pipe inspection (part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification,” *Automation in Construction*, vol. 73, pp. 1–11, 2017.

[46] D. Alejo, F. Caballero, and L. Merino, “Rgbd-based robot localization in sewer networks,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4070–4076.

[47] ——, “A robust localization system for inspection robots in sewer networks,” *Sensors*, vol. 19, no. 22, 2019.

[48] L. Merino, D. Alejo, S. Martinez-Rozas, and F. Caballero, “A rgbd-based system for real-time robotic defects detection on sewer networks,” in *Robot 2019: Fourth Iberian Robotics Conference*, M. F. Silva, J. Luís Lima, L. P. Reis, A. Sanfeliu, and D. Tardioli, Eds. Cham: Springer International Publishing, 2020, pp. 593–605.

[49] P. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.

[50] J. B. Haurum, M. M. J. Allahham., M. S. Lynge., K. S. Henriksen, I. A. Nikolov., and T. B. Moeslund., “Sewer defect classification using synthetic point clouds,” in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2021, pp. 891–900.

[51] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, “Generating synthetic point clouds of sewer networks: An initial investigation,” in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2020, pp. 364–373.

[52] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, “Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known cnn classifier,” *IEEE Access*, vol. 9, pp. 59 498–59 507, 2021.

[53] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, “Structured domain randomization: Bridging the reality gap by context-aware synthetic data,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7249–7255.

[54] pmd, “Development kit brief camboard pico flexx,” accessed: 13/3-2022. [Online]. Available: https://pmdtec.com/picofamily/wp-content/uploads/2018/03/PMD_DevKit_Brief_CB_pico_flexx_CE_V0218-1.pdf

[55] J. B. Haurum and T. B. Moeslund, “Sewer-ml: A multi-label sewer defect classification dataset and benchmark,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

Chapter 6

Conclusion

This Ph.D. covers work conducted from 2019 to 2022, focused on the automation of sewer inspection using image and point cloud data. The PhD was conducted as a part of the Automated Sewer Inspection Robot research project, focused on developing an autonomous robot for continuous sewer inspection.

The field of automated sewer inspections is a prime example of an inter-disciplinary and applied use of computer vision, and may not be the most well known application, but can have a major effect on society if developed sufficiently. While the methods applied in this field are not as advanced as in the general computer vision field, the methodologies are catching up to the historical lag, with more cutting edge algorithms being utilized and developed. This is partly because of a shift in how the task is approached, adopting the evaluation protocols used in the general computer vision field, but also due to the object recognition subfield of computer vision slowing down over the years after seeing massive progress over a short time period.

Within the field of image-based automation of sewer inspection we have investigated and detailed the history of the research field, documenting fundamental trends in algorithmic methodologies and evaluation protocols. Through this study we found that the image-based automation of the sewer inspection field consistently lagged behind the general computer vision field by years, and that there were three major hindrances causing this lag: a lack of open source code and data, and a lack of a common evaluation protocol. To this end, we proposed the world's first publicly available sewer defect classification dataset, Sewer-ML, consisting of 1.3 million images from Danish sewer inspections, annotated by professional sewer inspectors. Using this dataset we conducted an extensive benchmarking of state-of-the-art multi-label algorithms, finding that the claim of the sewer defect classification task being solved to be false. With the Sewer-ML dataset we further advanced the sewer defect classification task by introducing a novel multi-scale extension to the Hybrid Vision Transformer together with a clustering-based Sinkhorn tokenizer, improving upon the original benchmark algorithm by 6.32 and 5.00 percentage points on the validation and test splits, respectively. The Sewer-ML dataset also enabled the study of how well modern computer vision methods could be utilized to determine the water level in sewer pipes. Through this we validated the effectiveness of using the newest water level severity labels [1]. Lastly, A multi-task classification network was developed to not just predict sewer defects, but also infer the equally important sewer pipe properties and water level, leading to a performance increase of up to 6.34, 29.62, and 21.64 percentage points on the sewer defect, pipe shape, and pipe material classification tasks, respectively. This was achieved using a novel Cross-Task Graph Neural Network, which utilized the conditional probabilities across all the different classes. Through these developed methods we showed the effectiveness of drawing on and advancing the cutting edge research from the general computer vision and machine learning fields, while infusing the models with domain-based knowledge.

In the point cloud-based automation of sewer inspection field, we developed the world's first synthetic data generator for the sewer pipe domain. The synthetic data was empirically validated by comparison to real life data, and found to be a truthful representation. Using an updated version of the developed synthetic data generator, we constructed the world's first sewer point cloud dataset, consisting of both synthetic and real laboratory point clouds, and released it publicly. Using this dataset we benchmarked two commonly used geometric deep learning algorithms under different data scenarios, verifying the effectiveness of using synthetic point clouds to bootstrap the training process in the lack of historic data from sewer inspections.

The main contributions of this Ph.D. thesis can be summarized as follows:

- A detailed survey of the image-based automation of sewer inspections domains covering the last three decades. This survey uncovered the three main hindrances limiting progress in the field.

- The development of the Sewer-ML dataset, the world's first image-based sewer defect multi-label classification dataset, consisting of 1.3 million images from more than 75 thousand Danish sewer inspections. Sewer-ML was made publicly available, to help democratize the field.

- The development of two domain influenced evaluation metrics for sewer defect classification, $\text{F1}_{\text{Normal}}$ and F2_{CIW}, which directly incorporate the economic impact of the sewer defect categories into the model evaluation.

- The development of a Structured Domain Randomization based synthetic sewer point cloud generator, and the world's first point cloud-based multi-class classification dataset, containing over 17 thousand synthetic and real point clouds from sewer pipes. The synthetic data generator and dataset were made publicly available to help grow the point cloud based automation field.

- Benchmarking of commonly used computer vision algorithms on the tasks of image and point cloud based sewer defect classification, and image-based water level classification.
- A novel Multi-Task Classification network for simultaneously predicting sewer defect categories, water level, pipe shape, and pipe material. The developed method was based on the underlying conditional probabilities of different classes occurring together, estimated empirically, and allowed for incorporating heuristic rules.
- The development of a novel multi-scale Hybrid Vision Transformer for the sewer defect classification problem. The model was tested on the Sewer-ML, and shown to improve performance and improve efficiency of the model.
- State-of-the-Art on the Sewer-ML sewer defect classification with a $\mathrm{F2}_{\mathrm{CIW}}$ score of 60.57% using CT-GNN and competitive $\mathrm{F1}_{\mathrm{Normal}}$ score of 92.30% using MSHViT.
- State-of-the-Art on the Sewer-ML water level and classification tasks with a macro-F1 score of 71.30% and 81.10%, respectively, using CT-GNN.
- A competitive performance on the Sewer-ML pipe material classification task, with a macro-F1 score of 73.95% using CT-GNN.

In the future, there are several promising directions from within the general computer vision field which could have an immediate effect in the field of automation of sewer inspections. This especially concerns the task of obtaining more detailed and complete sewer inspections through automatic measures, and areas of data generation and labeling.

The automated sewer inspection field has historically, and currently, focused on automatically determining the defect category and location of the defect, ignoring the equally important aspects of defect severity level and type indicators, and the sewer pipe properties. While this has begun to change with recent works showing clear advances within water level and pipe property classification [2–6], it is clear that there is still progress to be made in order to generate a more complete and detailed sewer inspection. Specifically, the area of defect severity level and type indicators classification has been neglected, except for few very recent advances that attempt predicting the severity level for a subset of classes [7–9]. This is due to the increased fine-grainedness, long-tailed nature, and hierarchical structure of the data when these aspects of data has to be considered. However, many of these aspects has been the focus of the Fine-Grained Visual Categorization (FGVC) community for the last decade [10], resulting in a large set of methodologies that have been developed for exactly these scenarios. Therefore, a clear future research direction would be to apply the FGVC methods and adapt them with the relevant domain knowledge. This would be realizable by expanding the Sewer-ML dataset with the remaining labels from the original sewer inspections, as well as extending the current methods to use the vastness of metadata available to the water utility companies, such as the topology of the sewer network, the age of the pipes, the condition of neighbor pipes, and the geographic location, some of which are currently used to model sewer deterioration [11, 12].

As discussed throughout this thesis, a major hurdle with the automated sewer inspection field is the lack of publicly available annotated data. While we have reduced this hurdle for image classification task with the Sewer-ML dataset, the problem persists when investigating more semantically rich tasks such as defect detection and segmentation. Simultaneously, it is unrealistic to assume an immediate influx of publicly available sewer datasets, as this would require a complete shift in the traditions and culture of the research field, where commercial interests limits the possibility of freely sharing data. Therefore, there are three directions from the general computer vision field that could be used with great effect to advance the automated sewer inspection field.

Firstly, the use of synthetic data has been shown to be a viable way to not just get a large amount of diverse data but also superhuman level annotations [13, 14]. These approaches have led to advances such as dense facial landmark localization [14], large scale dataset generation for the multi-object tracking (MOT) [15] and autonomous vehicle domains [16, 17], and an increase of training data within niche areas such as fish farming [18] and wildlife monitoring [19]. Initial investigations into using synthetic data in the sewer domain have also shown to be promising [20–24]. The drawback of the synthetic data approach is the need of a sufficiently sophisticated data generator, which can capture the underlying complexities and variability of the domain. The general computer vision domain has had the luxury of leveraging the fact that many assets from the game and movie industry have been reusable for the relevant tasks. However, this is not the case for the sewer domain, where all assets have to be man-made through scanning techniques or manual modeling.

Secondly, one could begin annotating the Sewer-ML dataset such that it can be used for the defect detection and segmentation tasks. This is, however, an immensely time consuming and expensive endeavor to begin as not only is the annotation process often tedious, but the sewer inspection domain is highly specialized requiring the annotators to be trained to ensure that the defect categories are correctly annotated. These concerns are not confined to the automated sewer inspection field, but can be found across the entire computer vision and machine learning fields in general. Therefore, tools and procedures have been developed to help partially automate the labeling process. One such approach would be to manually label a small subset of the data, fine-tune a pretrained network on the annotated subset, and then use the predicted labels for a subset of the remaining data as a starting point. These annotations can then be refined to correct for errors made by the network, and the network can be fine-tuned anew, repeating this cycle until the entire dataset has been annotated. This was demonstrated in the creation of the MOTS dataset [25], where the bounding box annotations from the MOT datasets were lifted to instance segmentation annotations.

Thirdly, it may not be worth annotating all of the data. The machine learning fields focusing of alternative learning paradigms such as weakly-, semi-, and self-supervised learning have grown immensely within recent years, reaching impressive performances using little to no annotated data [26–33]. This is possible, as these approaches circumvent the need for large amount of annotated data, instead building on concepts such as contrastive learning [29, 30, 32, 34, 35], where heavily augmented versions of the same input are forced to have similar latent embeddings and dissimilar embeddings when the inputs are not identical, through the use of pseudo labels to enforce consistency between classification prediction of two augmented views of the same input [31, 36–41] , or by leveraging less semantically rich information such as only having the desired class for an object detection or segmentation task [29, 42–45]. These learning paradigms would be ideal for the automated sewer inspection domains with coarse level expert annotations, and a large amount of data which only grows larger each day. Some of these methodologies have also already been demonstrated for sewer defect detection [24, 46] and water level classification [4].

To conclude, within the last few years there have been made major strides within the field of automation of sewer inspections. We hope that in the future there will be an influx of computer vision scientists who will recognize the unique challenges of the field, and help advance this field by developing novel computer vision algorithms leveraging the domain knowledge. With these advances we are getting closer to automating parts of the inspection process. By increasing the inspection rate and enabling better asset management, we will build an even stronger and more robust infrastructure, with significant positive economical impacts in our modern society, as well as environmental and health-wise impacts by enabling earlier detection of faulty pipes.

References

[1] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 7th ed. Dansk Vand og Spildevandsforening (DANVA), 2015.

[2] J. B. Haurum, M. Madadi, S. Escalera, and T. B. Moeslund, "Multi-task classification of sewer pipe defects and properties using a cross-task graph neural network decoder," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2022, pp. 2806–2817.

[3] J. B. Haurum, C. H. Bahnsen, M. Pedersen, and T. B. Moeslund, "Water level estimation in sewer pipes using deep convolutional neural networks," *Water*, vol. 12, no. 12, 2020.

[4] F. Plana Rius, M. P. Philipsen, J. M. Mirats Tur, T. B. Moeslund, C. Angulo Bahón, and M. Casas, "Autoencoders for semi-supervised water level modeling in sewer pipes with sparse labeled data," *Water*, vol. 14, no. 3, 2022.

[5] H. W. Ji, S. S. Yoo, B.-J. Lee, D. D. Koo, and J.-H. Kang, "Measurement of wastewater discharge in sewer pipes using image analysis," *Water*, vol. 12, no. 6, 2020.

[6] H. W. Ji, S. S. Yoo, D. D. Koo, and J.-H. Kang, “Determination of internal elevation fluctuation from cctv footage of sanitary sewers using deep learning,” *Water*, vol. 13, no. 4, 2021.

[7] M. Wang, H. Luo, and J. C. Cheng, “Towards an automated condition assessment framework of underground sewer pipes based on closed-circuit television (cctv) images,” *Tunnelling and Underground Space Technology*, vol. 110, p. 103840, 2021.

[8] Q. Zhou, Z. Situ, S. Teng, H. Liu, W. Chen, and G. Chen, “Automatic sewer defect detection and severity quantification based on pixel-level semantic segmentation,” *Tunnelling and Underground Space Technology*, vol. 123, p. 104403, 2022.

[9] L. M. Dang, H. Wang, Y. Li, T. N. Nguyen, and H. Moon, “Defecttr: End-to-end defect detection for sewage networks using a transformer,” *Construction and Building Materials*, vol. 325, p. 126584, 2022.

[10] X.-S. Wei, Y.-Z. Song, O. Mac Aodha, J. Wu, Y. Peng, J. Tang, J. Yang, and S. Belongie, “Fine-grained image analysis with deep learning: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[11] B. D. Hansen, D. Getreuer Jensen, S. H. Rasmussen, J. Tamouk, M. Uggerby, and T. B. Moeslund, “General sewer deterioration model using random forest,” in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 834–841.

[12] B. D. Hansen, S. H. Rasmussen, M. Uggerby, T. B. Moeslund, and D. G. Jensen, “Comprehensive feature analysis for sewer deterioration modeling,” *Water*, vol. 13, no. 6, 2021. [Online]. Available: https://www.mdpi.com/2073-4441/13/6/819

[13] K. Greff, F. Belletti, L. Beyer, C. Doersch, Y. Du, D. Duckworth, D. J. Fleet, D. Gnanapragasam, F. Golemo, C. Herrmann, T. Kipf, A. Kundu, D. Lagun, I. Laradji, H.-T. D. Liu, H. Meyer, Y. Miao, D. Nowrouzezahrai, C. Oztireli, E. Pot, N. Radwan, D. Rebain, S. Sabour, M. S. M. Sajjadi, M. Sela, V. Sitzmann, A. Stone, D. Sun, S. Vora, Z. Wang, T. Wu, K. M. Yi, F. Zhong, and A. Tagliasacchi, “Kubric: a scalable dataset generator,” 2021.

[14] E. Wood, T. Baltrušaitis, C. Hewitt, S. Dziadzio, T. J. Cashman, and J. Shotton, “Fake it till you make it: Face analysis in the wild using synthetic data alone,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 3681–3691.

[15] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, “Motsynth: How can synthetic data help pedestrian detection and tracking?” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 10 849–10 859.

[16] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.

[17] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, “Structured domain randomization: Bridging the reality gap by context-aware synthetic data,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7249–7255.

[18] Y. Ishiwaka, X. S. Zeng, M. L. Eastman, S. Kakazu, S. Gross, R. Mizutani, and M. Nakada, "Foids: Bio-inspired fish simulation for generating synthetic datasets," *ACM Trans. Graph.*, vol. 40, no. 6, dec 2021.

[19] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, N. Joshi, M. Meister, and P. Perona, "Synthetic examples improve generalization for rare classes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[20] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, and J. Dong, "A multi-defect detection system for sewer pipelines based on stylegan-sdm and fusion cnn," *Construction and Building Materials*, vol. 312, p. 125385, 2021.

[21] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, "Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known cnn classifier," *IEEE Access*, vol. 9, pp. 59 498–59 507, 2021.

[22] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, "Generating synthetic point clouds of sewer networks: An initial investigation," in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2020, pp. 364–373.

[23] J. B. Haurum, M. M. J. Allahham., M. S. Lynge., K. S. Henriksen, I. A. Nikolov., and T. B. Moeslund., "Sewer defect classification using synthetic point clouds," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2021, pp. 891–900.

[24] C. Siu, M. Wang, and J. C. Cheng, "A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection," *Automation in Construction*, vol. 137, p. 104213, 2022.

[25] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *arXiv:2111.06377*, 2021.

[27] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "ibot: Image bert pre-training with online tokenizer," *International Conference on Learning Representations (ICLR)*, 2022.

[28] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent - a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 271–21 284.

[29] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9650–9660.

[30] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 1597–1607.

[31] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 22 243–22 255.

[32] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[33] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 15 750–15 758.

[34] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9640–9649.

[35] P. Goyal, Q. Duval, I. Seessel, M. Caron, I. Misra, L. Sagun, A. Joulin, and P. Bojanowski, "Vision models are more robust and fair when pretrained on uncurated images without supervision," 2022.

[36] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[37] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring," in *International Conference on Learning Representations*, 2020.

[38] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021.

[39] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 596–608.

[40] J. Li, C. Xiong, and S. C. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9475–9484.

[41] M. Zheng, S. You, L. Huang, F. Wang, C. Qian, and C. Xu, "Simmatch: Semi-supervised learning with similarity matching," *arXiv preprint arXiv:2203.06915*, 2022.

[42] Y. Li, Y. Duan, Z. Kuang, Y. Chen, W. Zhang, and X. Li, "Uncertainty estimation via response scaling for pseudo-mask noise mitigation in weakly-supervised semantic segmentation," *arXiv preprint arXiv:2112.07431*, 2021.

[43] S. Liu, K. Liu, W. Zhu, Y. Shen, and C. Fernandez-Granda, "Adaptive early-learning correction for segmentation from noisy annotations," *CoRR*, vol. abs/2110.03740, 2021.

[44] Z. Huang, Y. Zou, B. Kumar, and D. Huang, "Comprehensive attention self-distillation for weakly-supervised object detection," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[45] S. Jo and I.-J. Yu, "Puzzle-cam: Improved localization via matching partial and full features," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 639–643.

[46] S. Yang, Z. Zhao, Q. Yang, and J. Wang, "Attention guided image enhancement network for sewer pipes defect detection," in *2021 4th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, 2021, pp. 109–113.

Part II

Foundations for Image-based Automation of Sewer Inspections

Paper A

A Survey on Image-Based Automation of CCTV and SSET Sewer Inspections

Joakim Bruslund Haurum and Thomas B. Moeslund

The paper has been published in the
Automation in Construction Vol. 111, p. 103061, 2020.

The layout has been revised.

Abstract

This survey presents an in-depth overview of the last 25 years of research within the field of image-based automation of Closed-Circuit Television (CCTV) and Sewer Scanner and Evaluation Technology (SSET) sewer inspection. The survey investigates both the algorithmic pipeline, and the datasets and corresponding evaluation protocols. As a result of the in-depth survey, several trends within the research field are revealed, discussed, and future research directions are proposed. Based on the conducted survey, we put forth a set of three recommendations, which we believe will further improve and open the research field, as well as make the future research more reproducible: 1) The introduction of free and public benchmark datasets, 2) Standardized evaluation metrics, and 3) Open-sourcing the associated code.

1 Introduction

The sewers are often referred to as the “hidden infrastructure”, as it is typically out of sight and often hard to get to even though it is one of the most important elements in our modern societies. Without proper sewerage infrastructure society would be exposed to both environmental and health issues. There is therefore a large responsibility laid upon the maintenance and development of proper sewerage infrastructure worldwide, in both developed and developing countries. For instance, in the United States (US), there is currently more than 1.28 million km of public sewers and more than 800 million km of private lateral connections servicing approximately 240 million Americans. The sewerage infrastructure is expected to be further expanded by 2032, as an additional 56 million users will need to be connected, requiring a 271 billion dollar investment [1]. Comparatively, approximately 500 million km of public sewerage infrastructure served 444 million Chinese citizens living in urban areas of China as of 2014 [2]. This corresponds to approximately 5 m of public sewer per US citizen, but only 1 m of sewer per Chinese citizen. In order to accommodate the growing middle class and continuing urbanization in China, there is a clear need of expanding the sewerage infrastructure so the relative size of the infrastructure at least matches that of the American sewerage infrastructure.

However, it is currently difficult to meet increasing demands and ensure the quality of the sewers as defects go unnoticed. This is reflected by the 2017 American Society of Civil Engineers (ASCE) infrastructure report card [1]. In this report the ASCE gave a “D+” to the US wastewater infrastructure, citing the need of expanding the existing infrastructure, as the United States Environmental Protection Agency (US EPA) estimate 23,000-75,000 unintentional overflows in sewer systems a year. In China, there are currently several major problems such as the sewer pipes suffering from high infiltration and corrosion rate due to poor quality of the sewer construction. This is especially worrying, as 75% of the sewerage infrastructure is less than 15 years old.

Furthermore, the awareness and utilization of trenchless rehabilitation options is very poor, due to a lack of properly trained professionals [2]. This is problematic as inability to meet demands and poor maintenance of the infrastructure can lead to potential hazards such as environmental pollution, sinkholes, and an increase in transmission of diseases. There is therefore a clear global need of improving the construction quality, rehabilitation, and capability of the sewerage infrastructure.

As demands on the sewerage infrastructure continue to increase, so do the demands with regard to its inspection. Dirksen *et al.* [3] and Van der Steen *et al.* [4] have investigated the quality of human inspections, based on data from Germany, the Netherlands, France, and Austria. Dirksen *et al.* found that human inspectors do not identify defects 25% of the time while Van der Steen *et al.* found that the highly detailed EU standard (EN 13508-2) has led to an increase in incorrectly described features in the sewers. Automating the sewer inspection process is therefore of great interest in order to enable speeding up inspection times, reducing monetary costs, and removing potential human biases.

Automated sewer inspection encompasses a wide variety of technical fields; consequently, a lot of research has been conducted, resulting in a plethora of research papers. In order to keep track of the progress, several reviews and surveys have been conducted over the years. This includes studies into different kind of inspection technologies [5–10], robotic platforms for in-pipe inspections [11], automated inspection for concrete based infrastructure (including concrete pipes) [12], and the quality of human inspections [3, 4]. There has, however, not been a comprehensive review of image-based automated sewer inspection methods, except by Moradi *et al.* [13], who conducted a review of general trends in the field throughout the last 20 years. The Moradi *et al.* review does, however, not provide a detailed overview of the actual methodologies applied or evaluation protocols employed throughout the history of the literature. We believe a thorough overview and review of these methods and evaluation protocols is beneficial, as the research field has matured over the years. Specifically, methods working solely with images are of great interest, as sewer inspections are traditionally performed using a single camera.

Furthermore, the research area has been quite opaque throughout the years as everyone utilizes their own datasets and rarely publicly share code or data. This is often due to the ownership of the data belonging to a third party such as water utility companies, or that the research is conducted in collaboration with industry partners. This makes it hard to directly compare the performance of different methods, and readily advance the field, as seen in the other computer vision fields like image classification, object detection, and object segmentation fields where public data and shared code have been drivers.

The contributions of this article are as follows. We have created a thorough overview of image-based algorithms employed for classification, detection, and/or segmentation of defects in sewers based solely on image or video data. We have similarly investigated and created an overview of the utilized datasets and evaluation protocols of each methodology. Based on these overviews, we analyze the trends and tendencies throughout the literature and history of the field. Lastly we provide a set of recommendations on how to improve the evaluation protocols, as to better foster the research field of image-based automated sewer inspections.

The rest of the paper is organized as follows: Section 2 provides an overview of different sewer inspection technologies. Section 3 describes the methodology used in the survey. Section 4 provides a comprehensive overview of image-based defect detection, segmentation, and classification algorithms. Section 5 describes the datasets and evaluation protocols employed throughout the literature. Conclusions and recommendations for future research within this field are presented in Section 6.

2 Sewer Inspection Technologies

Most sewer inspections are conducted using a single camera. This technique is denoted Closed-Circuit Television (CCTV) and has been used for more than 40 years [14]. The camera is either mounted on a remotely controlled tractor, a manually pushed rod, or placed on the edge of the sewer opening with a zoom lens. In either case, an operator controls the camera/tractor and either inspects the sewer on site or back at the office. In some cases, one or more fish-eye lenses are used to capture the full area around the tractor. A commonly used system that utilize fish-eye lenses is the IBAK PANORAMO [15]. While the tractor-based approach is often utilized for large pipes, resulting in relative smooth motions and centered footage, the push rod approach is most often used when expecting laterals and small pipes, where the tractor cannot gain access. The footage obtained with push rod cameras is, however, not centered and contains erratic camera motion, which directly detracts from the quality of the obtained data.

The Sewer Scanner and Evaluation Technology (SSET) was commercialized in 2001 [16]. The SSET device contains a suite of sensors including both a standard and a fish-eye CCTV feed combined with inclinometers to determine the inclination of the device. SSET automatically pre-processes the fish-eye data in order to provide a single image containing the entire pipe wall, which is used as the main output. Similar techniques have been used with the PANORAMO.

Other visual methods include thermal imagery and laser profiling, which are used to detect thermal anomalies and generate 3D profiles of pipes, respectively. Non-visual methods have also been used such as electromagnetic, acoustic, and ultrasound based methods.

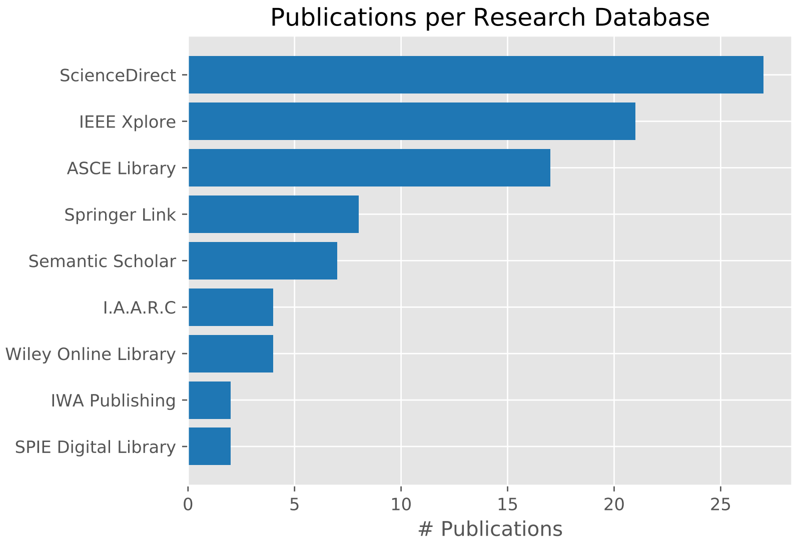


Fig. A.1: Bar plot of the amount of papers investigated research databases with more than one relevant paper.

For an in-depth description and comparison of these methodologies, we refer to Duran *et al*. [7] and Liu and Kleiner [10].

Lastly, there has been a long term research interest in developing autonomous robots for pipe inspections, such as the MAKRO [17], PIRAT [18], KANTARO [19], and SIAR [20] robots. These robots are typically fitted with a large suite of different sensors, allowing for different kinds of analysis. For an in-depth survey on different pipe inspection robots, we refer to the survey by Tur and Garthwaite [11].

3 Survey Methodology

This survey is based on a thorough investigation of the academic work that has been applied to CCTV and SSET based sewer inspections. We have investigated papers from 1994 to 2020 which exclusively use image data and not technologies such as laser profiling or ultrasound. Only papers that are focused specifically on defect recognition and sewer inspection have been included. Therefore, works such as the image quality estimation of Yang *et al*. [21, 22] and the estimation of extrinsic camera parameters of Cooper *et al*. [23–25] are not included. Similarly, commercial applications such as the “Deep Learning Sewer Defects Detection” cloud service from InLoc Robotics or the robot solutions offered by RedZone Robotics are only included if peer-reviewed papers are available. The papers have been selected by searching several large research databases (IEEE Xplore, Springer Link, ScienceDirect, ASCE Library, IWA Publishing,

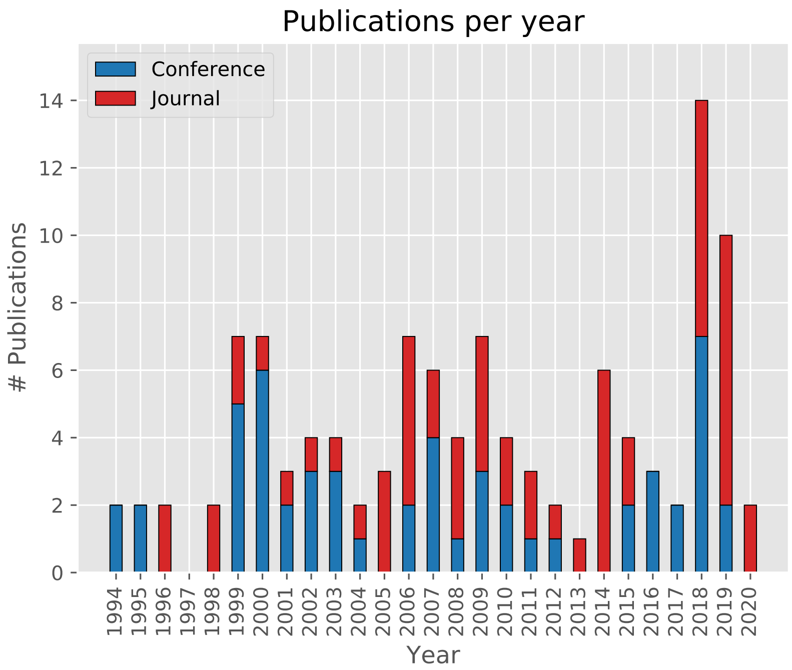


Fig. A.2: Stacked bar plot of the amount of conference and journal papers per year from 1994 to 2020.

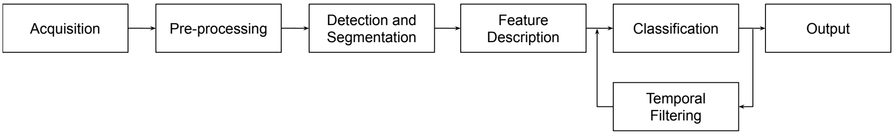


Fig. A.3: Illustration of the generalized pipeline.

Scopus, and Web of Science), and by recursively goings through previous referenced work. Figure A.1 shows the distribution of papers across research databases with more than one relevant paper. Conference proceedings and journals in English have been considered. If a paper was not immediately accessible, we have made efforts in order to retrieve the paper through official university channels. In total, 113 papers were investigated, 54 from conference proceedings and 59 from journals. In Figure A.2, the distribution of conference and journal papers from 1994 until 2020 is shown. All surveyed papers are listed in Table A.3 in A.A.

4 Image-based Automated Inspection

To provide a framework for analyzing the different investigated methods, we shall use a general computer vision pipeline, illustrated in Figure A.3 consisting of the following steps: **Acquisition**, **Pre-processing**, **Detection and Segmentation**, **Feature Description**, **Classification**, and **Temporal Filtering**. This is not to say that all systems use all these steps, but the framework provides a concise and unified way of describing and comparing different systems. In the following, each step of the generalized pipeline will be surveyed.

Table A.1 contains an overview of the methods described throughout the survey. As several of the methods have been developed over a series of papers, only the most representative paper is included in the table. In case an author group has produced several different methods, each method is shown in the table, and the author group affiliation is indicated. In cases where *e.g.* several classifiers are tested, only the best performing method is shown.

4.1 Acquisition

Data acquisition has been achieved through two general inspection technologies: CCTV [21, 26–91] and SSET [16, 92–114]. It should be noted that Wu *et al.* [114] actually use the front camera of the SSET device unlike the other mentioned papers, which use the unwrapped pipe wall images. In a few cases, CCTV footage has been recorded with a fish-eye lens that has either been unwrapped in order to obtain an image of the entire pipe wall, similar to SSET [115–123], or simply used the original wide-angle images [45, 124–135]. A zoom CCTV camera has also been utilized [136].

4.2 Pre-processing

The pre-processing methodologies vary widely depending on the applied algorithm. However, some methods, which are not simple rescaling or normalization, reoccur in several papers. We do not consider data augmentation to be a pre-processing step but rather a dataset related procedure.

Color Space

All modern inspection systems record video in color; but, many algorithms do not operate directly in the RGB color space. Several systems convert the RGB images to grayscale by averaging the color channels [33, 58, 59, 63, 64, 67–74, 92, 94, 95, 117, 124]. Weighted averages such as the NTSC standard [135], brightness [126, 127], luminance [62, 114], based on the pipe material [116], or a linear transformation based on Fischer's linear discriminant classifier [96, 100, 102, 105, 109, 110] have also been utilized when emphasis on specific color combinations was necessary. Color conversion has also been used in conjunction with image enhancement [111–113]. In several cases, grayscale images are used, but it is not stated whether or how the conversion is performed (though an equally weighted average is assumed) [16, 21, 26–32, 34–53, 55, 57, 65, 66, 75, 76, 93, 97–99, 101, 103, 104, 106–108, 115, 125, 128, 129, 133]. The YCbCr [122], HSV [56, 84, 88, 90], HSB [117, 118], and RGB color spaces [54, 60, 61, 76–83, 85–87, 89, 91, 118–121, 123, 130–132, 134, 136] are also utilized.

Noise Removal

A general pre-processing step in image processing, is the use of noise removal filtering in order to remove spurious or high frequency noise. This has typically been through mean [42, 56, 57, 122], median [31, 36, 52, 53, 57, 59, 61, 111–113, 119, 120, 122, 135], or Gaussian filtering [75, 76, 121, 126, 127, 132]. In some cases, special noise removal approaches are applied. Kirstein *et al.* [116] utilize an anisotropic Gaussian filter to remove directional noise while Müller and Fischer [115] use an FFT based noise removal and Fourier amplitude modulation approach. In some works unspecified noise removal is applied [64, 131].

Image Stitching, Mosaicking, and Unwrapping

In most cases, when an unwrapped wide angle CCTV image is used, the unwrapping is performed directly within the recording device. In a few cases, the unwrapping has, however, been part of the algorithmic design, where it has been improved [32, 122, 123, 126, 127]. In these cases, the images are often also mosaicked and stitched together using custom made approaches. Künzel *et al.* [122] stitch the unfolded images by minimizing the absolute difference between the two image regions, while penalizing non-optimal transitions between consecutive images, in order to obtain the optimal seam. The obtained seams are blended using Poisson Blending, in order to smooth any inaccuracies, while maintaining the gradient information. Piciarelli *et al.* [123] construct a mosaick using an iterative methodology, where a set of features are extracted from the current mosaick and the unwrapped image, and matched after outliers have been removed. The remaining features are used to construct a homography between the mosaick and unwrapped image, resulting in an updated mosaick.

Image Enhancement

Lastly, before further processing, enhancement methods are applied in several papers. Simple approaches utilize histogram equalization [31, 55, 56, 130, 131], contrast stretching [55, 90, 122], and (Laplacian) sharpening [55, 64, 130].

McKim and Sinha propose a local enhancement method in order to enforce constant background illumination [96, 98], while Iyer and Sinha propose an enhancement method to increase contrast between dark pixels and a computed “median background image” [111–113]. Ruiz-del-Solar and Köppen [32] applies a contrast enhancement approach where areas with high contrast are amplified while low contrast areas are attenuated.

4.3 Detection and Segmentation

Detection and segmentation of regions of interests are a very broad topic with widely different approaches on how to arrive at intended result. We have identified a series of reoccurring elements, which have been analyzed. Some papers also utilize a more advanced combination of algorithms. We denote these algorithms as *compound approaches*.

Edge Detection

Edge detection is a key element in the detection of change in the image such as the transition between a defect on the pipe wall and the normal appearance of the pipe wall. Several approaches have been employed: the basic Sobel operator [51–53, 58, 65, 66, 126, 127], the Laplacian [36], the Laplacian of Gaussians (LoG) [111–113], and the Canny edge detector [26–28, 30, 31, 121, 125, 129]. The Canny edge detector is utilized in different ways such as only considering horizontal or vertical edges in a sliding window approach [115], introducing an extra filtering step based on size and shape [116], and using multiple threshold settings to adaptively find the "core edges" [132]. In some cases, an undefined operator is used [33, 37, 39, 55, 92].

Fieguth and Sinha [95–106] introduced a special edge detector for crack segmentation which combined two different edge detectors based on local patches and two neighboring patches. The detectors utilized the ratio of the patch means and cross-correlation of the patches, and the responses were combined using an associative symmetrical sum. This was repeated over several orientations of the patches. The crack segments were linked by assuming local linearity and combined using the Hough transform. This approach was expanded on by Sinha and Fieguth [107] using 3 different sized windows in the comparison step, only testing for cracks in four directions (0, 45, 90, 135 °), and using a nearest neighbor based linking process.

Thresholding

Thresholding is in general a simple but key step in determining the regions of interest in an image. The operation can be as simple as using a predetermined threshold [99, 100, 102, 104, 107, 111–113, 129, 135] or in a single case using two values for hysteresis thresholding [135].

Predetermined thresholds, however, requires setting a single static value based on prior knowledge of the problem. This is problematic in dynamic environments: consequently, Otsu's method is widely used [21, 33, 48, 49, 57, 59, 61, 63, 64, 97, 109, 110]. Iterative approaches are employed in order to handle the dynamic challenges [34, 35, 55, 56, 90]. K-means with 2 clusters has similarly been used as a non-parametric operation [62].

Alternative approaches utilize prior knowledge by assuming two peaks in the histogram, and setting the threshold point at the minimum [37] or median [36] in between the peaks. Lastly, several papers utilize thresholding without stating any specific threshold, though in most cases, a predetermined global threshold can be assumed [39, 40, 51–53, 92, 95, 96, 98, 101–103, 105, 106, 130].

Morphology

Grayscale morphology has been extensively utilized throughout the literature [39, 40, 48, 51–53, 90, 96–98, 100–103, 105, 107–113, 135]. The approach was introduced in the automated sewer inspection domain by Sinha *et al.*, who proposed an hierarchical approach utilizing grayscale opening, where a series of structuring elements (SEs) with increasing size were applied in order to segment out different defects in an SSET image [96, 98, 102, 105, 107–110]. The amount and size of the SEs were chosen based on a discriminative analysis. Yang and Su applied these approaches to CCTV videos [48] and compared the segmentation to a set of "ideal morphologies" as well as using the grayscale opening top-hat and closing bottom-hat operations [51–53]. Iyer and Sinha [111–113] have also utilized an ensemble of grayscale morphology operations for crack segmentation in conjunction with geodesic reconstruction and Laplacian curvature estimation.

Binary morphology is a commonplace methodology that is used for removing noise and holes in binary images [56, 57, 59, 61, 63–66, 90, 129, 135]. It has, however, also been employed as a core algorithmic step for defect detection and segmentation. Yang and Su proposed a morphological algorithm called Morphological Segmentation based on Edge Detection (MSED), which at times outperforms the grayscale morphological operations for defect segmentation [51–53]. Mashford *et al.* [118] proposed a generalization of the erosion operation, called α-erosion, in order to combat noise in the defect segmentation step. Halfawy and Hengmeechai [58] utilized a series of differently oriented SEs to extract cracks from all orientations.

Geometric Fitting

In several cases, the geometric knowledge of the sewer pipes and defects can be leveraged.

Least squares estimation is a commonly used approach to estimate the best fitting ellipse of a circular pipe. The extracted ellipse can be used to determine whether the pipe is broken or deformed, by fitting pixels [135], pixel groups [27], and in combination with frequency analysis [33].

Hough transformation has been utilized in similar ways as least squares estimation. The circular Hough transform was applied in order to extract laterals [33, 36, 97] and joints [26, 27]. The line Hough transform has been used for flow line detection [116], shape description [62], crack segment linking [97, 99, 103], and removal of text boxes in an edge image [58].

Several geometric approaches have been employed for estimating 3D properties. Kolesnik and Baratoff [36] utilized least squares estimation to construct a 3D understanding of inlets and the main pipe. Broadhurst *et al.* [28–30] utilized a method called Reflective Photometric Stereo based on reflected illumination instead of direct illumination to detect laterals and describe its 3D position and orientation. Swarnalatha *et al.* [55] estimated the depth of detected cracks utilizing angles of incidence and a priori knowledge of the pipe.

Camera Movement

The camera movement is utilized to detect time periods with potential defects, in order to reevaluate previous sewer inspections. This has been done by using optical flow [60, 61], or through constructing a time series of the mean absolute difference between consecutive frames and looking for time spans with low differences [57]. Chen *et al.* [86] utilized a bi-directional optical flow approach to filter out abnormal frames in the video sequences, which were caused by erratic camera movement and not by defects in the pipe.

Compound Approaches

Guo *et al.* applies a combination of an adaptive multi-resolution analysis step and expectation maximization based clustering [128], as well as K-means [130] to segment an input image. Similarly, Xue-Fei and Hua [56] test a clustering method called Quick Fuzzy C-Means (QFCM) for defect segmentation. The K-means and QFCM methods, however, suffer from requiring the number of clusters k to be set before processing.

Ruiz-del-Solar and Köppen [32] employs an approach called Simplified Boundary Contour System (SBCS). SBCS is stated to be a simplified approximation to the mammalian visual pathway. The approach is based on a set of Gabor filters and a set of "competitive" and "cooperative" stages in order to find a good segmentation of the image.

Taylor *et al.* [31] detect laterals by utilizing reflected illumination and the Canny edge detector. The laterals are detected by trying to fit circles to each edge based on the approach of Cooper *et al.* [25] and using a priori assumption of the curvature of the edge.

Paletta *et al.* [34, 35] extract a "support map" based on a priori knowledge of where pipe inlets are typically located. This is done by detecting the vanishing point of the sewer by iteratively thresholding the image until a certain amount of pixels are segmented. Based on the detected vanishing point, regions with high probability of inlets are found. This is based on the assumptions of inlets being to the left and right of the vanishing point within an angle of $\pm 30°$.

Moselhi and Shehab [37, 39, 40] use a combination of simple transformations such as complement, background subtraction, and gradient estimation. These transformations are combined based on which defects were to be extracted.

Browne *et al.* [41, 42] detect cracks by subdividing the image into smaller sections. This is achieved by dividing the outer edge of the image into 14 64×64 regions [42] or by dividing the entire image into 16×16 regions [41] before further analysis.

Ahrary *et al.* [126, 127] utilize sequential horizontal and vertical sliding windows where the patches are compared with an averaged normal prototype image through auto-correlation. This results in a similarity score for each patch, and used to determine the state of the patch.

Guo *et al.* [130, 131] propose a simple segmentation method based on thresholding the difference between an image of interest with a reference image of a normal pipe, which requires registering the images as a pre-processing step. Guo *et al.* [132] also propose a segmentation approach based on region growing using edges as guides to select the seed points.

Huynh *et al.* [65, 66] propose a crack segmentation method based on the skeletonization of an edge extracted image and scanning for linear or horizontal segments in the binary image.

Halfawy *et al.* [57, 61] segment a standard CCTV image into three components: pipe wall, water flow, and the end of sewer (EOS). The EOS is found based on row and column histogram data, and assuming a pre-defined radius of the center. The pipe wall and water flow are segmented by investigating the intensity along horizontal lines below the EOS, and drawing vertical line where it transitions from high to low intensity and back. The intersecting points result in lines sloping towards the center of the EOS. A set of sanity checks are used to verify that the two lines meet near the EOS and forms a valid triangle in which the water flow is contained. Defects can be detected by using Otsu's method and morphology to obtain a binary image where pixels are assigned to the pipe wall or water flow. If any pipe wall pixels are within the pre-defined water flow area, the pixels are extracted for further classification.

Hawari *et al.* [135] utilize a set of defect dependent segmentation algorithms. Sediments are segmented by utilizing Gabor filters and active contour segmentation, while least squares estimation is utilized to estimate pipe deformation. Displaced joints are segmented using the distance transform and a set of heuristic rules, and cracks were found through a combination of grayscale morphology and heuristic rules.

4.4 Feature Description

We split the investigated feature descriptors into three categories: Hand-crafted, semi hand-crafted, and designed descriptors. This is based on whether the exact features were either hand selected, a selection of statistics and mathematical properties, or more complex algorithmic designs, respectively. A fourth category also exists, where the features are learned directly through the applied classification method, but is not discussed in this section.

Hand-crafted

The hand-crafted descriptors consist of geometrical and intensity attributes that are interpretable and describe the object of interest.

Typical geometrical features of an object are area [65, 66, 109–113], length and width [55, 56, 90], or a variety of attributes based on shape, position, eccentricity, pixel intensities, etc. [31, 33, 37–40, 48, 49, 58, 62–64, 95–97, 100, 103–106, 108, 125, 135]. In several cases, several initial features are selected but then reduced through Principal Component Analysis (PCA) [48, 49] or other discriminative analysis [95–97, 100, 103]. Paletta *et al.* [34, 35] simply vectorize the input image and applied PCA.

Semi Hand-crafted

The majority of semi hand-crafted features build upon frequency domain analysis. Mashford *et al.* [119, 120] applied the discrete Haar wavelet transform to each color channel and represented each pixel by the concatenation of each color channel response in a 8×8 window. Similarly Browne *et al.* [42] utilized the Shannon Entropy of the normalized Haar wavelet coefficients of several scales, and Haar-like features have been used by Halfawy *et al.* [57, 61]. Browne *et al.* [41] similarly investigated using wavelet transforms to create a "space-frequency energy signature" based on the empirical cumulative distribution function of wavelet coefficients, which were further dimensionality reduced using PCA.

Yang and Su [46, 47, 50] have utilized the discrete wavelet transform that divides the image into a low-frequency and three high-frequency subimages. For each of the high-frequency subimages, four Gray-Level Co-occurrence Matrices (GLCMs) with a distance of 1 pixel in four directions (0, 45, 90, 135 °) are calculated. For each GLCM; the entropy, correlation, and cluster frequency of the matrix are calculated and averaged. The three GLCM features were decided upon through a thorough discriminant analysis of seven different GLCM features [46, 50]. Wu *et al.* [114] applied a similar approach using the Contourlet transform. This produces eight high-frequency subimages and one low-frequency subimage. The low-frequency subimage was filtered with the Maximum Response (MR) filter bank resulting in eight responses. For each image, a 64-dimensional feature vector was constructed consisting of the mean, standard deviation, skewness, and kurtosis of each of the 16 subimages. Wu *et al.* also conducted a thorough comparison on different feature vectors, however, all performed worse than the statistical features.

Statistical features have also been used by Ganegedara [121] and Ye *et al.* [88]. Ye *et al.* applied the "lateral Fourier transform" and the Daubechies wavelet transform (Db4). From these transforms, a set of statistics are calculated and concatenated with the first seven Hu invariant moments and other image statistics. Statistical features are similarly used on color information [117, 132].

Designed Descriptors

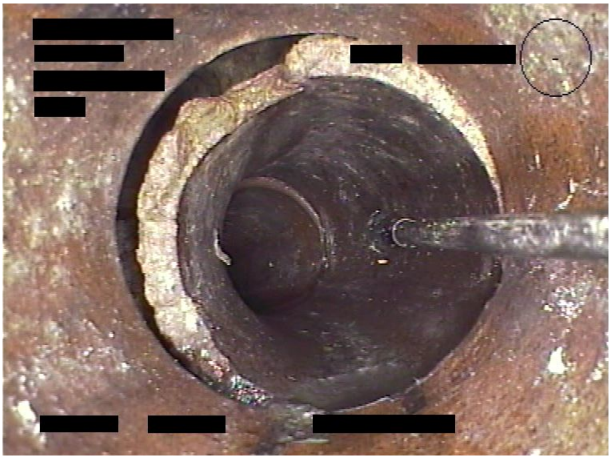
Histogram of Oriented Gradients (HOG) is applied by Halfawy and Hengmeechai [59] in order to detect root intrusion by analyzing segmented areas. The Scale Invariant Feature Transform (SIFT) has been applied by Guo *et al.* [133] to detect change between two frames and, thereby, detect defects. Moradi and Zayed extracted a spatio-temporal version of HOG [137] where the volume of frames is densely sampled in an overlapping cuboid grid, each of which are split into $2 \times 2 \times 2$ subregions. For each subregion, a normalized histogram of oriented gradients is extracted, and stored as a series [138]. Moradi *et al.* [76] has also looked into using histograms of SIFT features. The GIST feature descriptor has been utilized by Myrans *et al.* [67–74], who found that for their pipeline it performed better or equal to HOG and SIFT. Piciarelli *et al.* [123] calculated the Local Binary Patterns (LBP) of a large unwrapped CCTV image by utilizing a patch based approach.

4.5 Classification

The classification task can be split into three sub-tasks: binary (defect/no defect), multi-class, and multi-label classification. The different classification sub-tasks are demonstrated in Figure A.4. Two types of methods are applied to tackle these different tasks: classifiers based on predetermined heuristic rules, and data-driven classifiers that have learned a set of rules based on the supplied training data.

Heuristic Rules

Heuristic rules have been used in a large amount of papers often through thresholding based on geometrical attributes of segmented elements such as area [58, 65, 66, 102, 109–113], length and width [55, 56, 90], or a variety of different geometrical attributes [26–31, 61, 118, 135]. The features do, however, not have to be geometrical in nature for thresholding to be utilized. Other thresholded features include auto-correlation between two patches [125–127], L_2 distance between color



[redacted] [redacted] [redacted] [redacted] [redacted] [redacted]

Task	Prediction
Binary	Defect
Multi-class	Displaced
Multi-label	Displaced and Broken

Fig. A.4: Example of different levels of generalization of the classification task. A pipe with two defects, displacement and broken pipe wall, is shown, and below the predictions per generalization level are shown. Binary classification predicts whether there are any defects in the image. Multi-class classification predicts the most prominent defect. Multi-label classification predicts all defects present in the image.

histograms [132], the amount of SIFT keypoints matched using nearest neighbor assignment [133], entropy of a GLCM matrix [61], and the difference between two images [131]. Müller and Fischer [115] use a distance threshold based on multi-scale analysis and height of edges to classify connections and sockets in unwrapped CCTV images. Kirstein *et al.* [116] find the flow line in an image by optimizing a graph of line segments using Dijkstra's shortest path algorithm. Furthermore, areas of interest are identified based on time series data on mean absolute differences [57], optical flow information [60, 61, 86], and analysis of text on the video frame [84].

Machine Learning

Multi-Layer Perceptrons (MLPs) have been used extensively since the turn of the century. Standard MLPs have been utilized for the multi-class classification problem [38–40, 47, 96], and crack classification combined with finite element analysis to determine structural integrity [54]. MLPs have also been combined with fuzzy logic [92–96, 100, 103–106, 108], and a special MLP variation called Kohonen Self-Organizing Maps (SOM) has been employed for clustering different defects [121] and to detect sockets in pre-segmented image [32]. Browne *et al.* [41, 42] utilized logistic regression and compared it to using an MLP [41].

Radial Basis Function Networks (RBNs) are a special kind of MLP where the activation functions are radial basis functions, which have been used extensively by Yang *et al.* [21, 47–49] for multi-class classification. Paletta *et al.* [34, 35] utilize RBNs for inlet detections where the predictions are combined with a multi-resolution pyramid framework. For each resolution, the predicted probability of each investigated region from the RBN are fused in a Bayesian framework generating a "posterior map".

Convolutional Neural Networks (CNNs) have seen a spike in popularity in recent years as datasets and compute resources have increased. This has allowed end-to-end learning of a variety of powerful classifiers.

The first use of CNNs was by Browne *et al.* [43, 45, 124] in 2003, where they were applied on crack segmentation and the multi-class classification task of detecting the presence of sewer landmarks (joints and laterals). In 2004, Ouellette *et al.* [44] trained a CNN for the task of per-pixel classification in a crack segmentation context. The authors used a Genetic Algorithm (GA) for optimizing the weights of the network. Künzel *et al.* [122] applied a CNN for per-pixel classification of eight classes in a semantic segmentation context utilizing an adapted version of the Full-Resolution Residual Network (FRRN) architecture. Similarly, Wang and Cheng [82] applied a custom CNN utilizing multi-scale dilated convolutions [81] and dense Conditional Random Fields (CRFs) formulated as recurrent neural network layers, allowing for end-to-end learning. A VGG-16 model pre-trained on ImageNet [139] is used as the backbone of the CNN.

Cheng and Wang utilized a fine-tuned Faster R-CNN model with ZFNet as the backbone for detecting sewer defects [79, 80]. Kumar and Abraham [78] proposed using a two stage methodology, by first classifying images in a multi-class manner using a small CNN inspired by Kumar *et al.* [77], and subsequently a defect specific YOLOv3 model, if a defect was classified by the small CNN. Kumar *et al.* [83] further compared the Faster-RCNN model, first used by Cheng and Wang [79, 80], with the YOLOv3 and the Single-Shot Multibox Detector (SSD) models. Yin *et al.* [91] similarly applied the YOLOv3 model to detect defects in sewer networks.

However, the majority of CNNs have been employed for the general classification of an image, where one or more labels are associated with the image. Kumar *et al.* [77] proposed using an ensemble of binary CNNs for detecting three different defects in an image. The ensemble is trained in a one vs. all manner, which enables multi-label classification. This method was extended by Meijer *et al.* [134] using a single CNN for the task, instead of an ensemble.

Hassan *et al.* [85], Li *et al.* [89], Xie *et al.* [136], Tennakoon *et al.* [87], and Chen *et al.* [86] have utilized CNNs. Hassan *et al.* fine-tune the AlexNet architecture that is pre-trained on ImageNet in order to classify the pipe as one of six defects. The results were combined with the optical character recognition method of Dang *et al.* [84] in order to provide full inspection reports. Li *et al.* train a modified Residual CNN (ResNet) with 18 layers, where the conditional probability of each defect type, and the probability of the image containing no defects, are determined. This is done by internally computing the probability for the binary defect problem, and a probability for each defect (assuming a defect is present). The probability vectors are subsequently combined into a single probability output vector. Xie *et al.* utilize a two stage hierarchical approach by first using a small CNN trained for the binary classification problem. If a defect is detected, the image is passed through a second fine-tuned CNN in order to determine the defect type. Tennakoon *et al.* test a custom made CNN and a fine-tuned version of a ResNet with 50 residual layers (ResNet50) that was initially trained on ImageNet. Chen *et al.* utilize an approach similar to Xie *et al.* where a small, but fast, SqueezeNet is used to detect any abnormal frames, and a larger, but slower, InceptionV3 is used to classify the type of defect.

Moradi *et al.* [76] proposed a two stage hierarchical approach for crack detection. First, a Hidden Markov Model (HMM) is trained exclusively on images of normal pipes in order to detect anomalies. This has been tested using spatio-temporal HOG like features [75] or histograms of SIFT features [76]. For each identified defect image, a CNN is applied in order to determine whether it is a crack defect or not.

Fuzzy Classifiers are applied in various ways for defect classification. Several Neuro-Fuzzy (NF) methods are proposed where the input and/or output of an MLP are fuzzified. Sinha *et al.* [95, 96, 100, 103–105, 108] utilized an NF approach where continuous input features are fuzzified by the use of three membership functions. These membership functions convert the feature value into three linguistic representation ranging from Small, Medium and Large. The NF approach was also compared with, and outperformed, a fuzzy K-NN classifier, and a normal K-NN classifier. Chae *et al.* [16, 92–94] used an ensemble of MLPs to determine attributes of cracks, joints, and laterals in an image and applied a set of fuzzy rules to consolidate these rules into a condition rating for the pipe segment.

Chaki *et al.* [62] performed crack segmentation, by utilizing fuzzy multi-factorial analysis, and fuzzy logic is used to determine the severity of segmented cracks [63, 64] and to distinguish corrosion and pipe connections [119].

Support Vector Machines (SVMs) are used for several tasks. Mashford *et al.* [117–120] have utilized SVMs for binary per-pixel classification defect segmentation task. Myrans *et al.* [67, 73] use an SVM for the binary classification task. Halfawy and Hengmeechai similarly used an SVM for detecting intruding roots [59] as well as classifying the type of frame (looking forward, into pipe wall, info screen, or tilted view) using the SVM^{multiclass} library [61]. SVMs are also applied for multi-class defect classification by using an ensemble of SVMs by Yang and Su [47] and Ye *et al.* [88]. The ensemble is constructed based on the one vs. all paradigm [47], or the one vs. one paradigm [88].

A special case of one-class SVMs (OCSVMs) have likewise been investigated. OCSVMs are typically used for anomaly detection where normal data is abundant, but data with defects can be lacking. The OCSVM is trained using only the normal pipe data forcing all data points to be within a hypersphere. Myrans *et al.* [71] compared the performance of the OCSVM against a SVM and random forest binary classifier. Piciarelli *et al.* [123] utilized a OCSVM to generate a "heatmap" on an unwrapped CCTV image to indicate anomalous areas by processing the image in a sliding window manner.

Random Forests (RFs) have been utilized extensively by Myrans *et al.* in recent years for both the binary [68, 69, 72, 73] and multi-class classification tasks [70, 74]. Through their work, it was found that the Extremely Randomized Trees algorithm performed better than traditional RFs and SVMs. Binary detection was further improved by stacking the RF and SVM classifiers using an SVM stacking classifier. The multi-class classification problem was approached using a hierarchical structure, where first the binary RF classified whether images contained defects. If so, the images were analyzed by a specialized defect classification RF. Myrans *et al.* found that using an ensemble of binary RFs trained in a one vs. all manner performed better than a single multi-class RF or an ensemble of RFs trained in a one vs. one manner [74].

Boosting has so far been utilized sparingly for sewer defect detection. Sarshar *et al.* [57] employed AdaBoost to classify the frame type. Wu *et al.* [114] compared the performance of several ensemble methods and found that the RotBoost algorithm performed the best. RotBoost is a combination of the Rotation Forest, which in itself is an extension of the RF algorithm, and AdaBoost algorithms.

4.6 Temporal Filtering

Efforts have been towards temporally filtering the classification outputs in order to reduce the effect of noise and camera movement. Myrans *et al.* [69, 72, 74] used a HMM and Order Oblivious Filtering (OOF) for modeling the transition from normal to defective pipe segments. The approach can be used online and offline based on whether a backwards pass is performed through the predictions. The OOF further smooths the classification by assigning the state with a majority of occurrences over a symmetrical window. Pan *et al.* [26, 27] smoothed their tracking of pipe joints by using a simple criteria for the center and radius of the tracked circle, and the amount the parameters may vary in two consecutive frames. Paletta and Rome [35] consider different temporal smoothing approaches such as Bayesian fusion of the

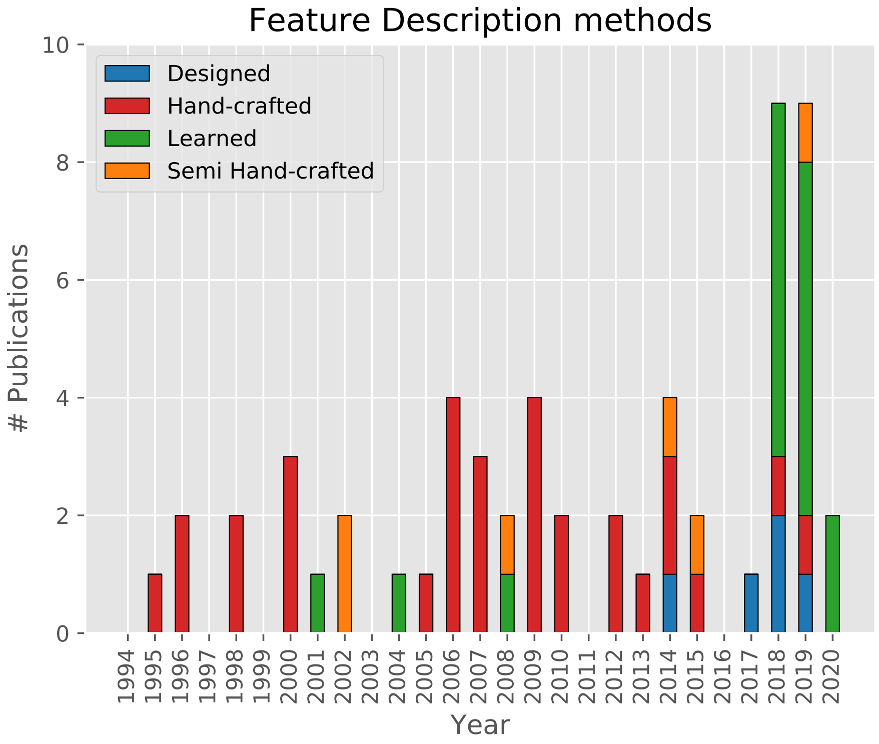


Fig. A.5: Distribution of the feature description methods used throughout the years. Based on the papers in Table A.1.

predictions or correlation based tracking of the detected inlets. A Q-learning based reinforcement learning approach is, however, used in order to actively fuse sensorimotor measurements and plan appropriate operations. Khalifa *et al.* [64] utilize temporal filtering methods to model the development of cracks over several months through a Markovian prediction framework.

4.7 Discussion and Future Directions

In the previous, subsection the different stages of the automated inspection pipeline have been presented. It is clear from the investigated literature that there have been several significant algorithmic trends in the last 25 years.

In the acquisition, stage there has been a clear trend of primarily using standard CCTV images, whereas SSET and unwrapped CCTV images have only been utilized by a few author groups. This makes a great deal of sense as CCTV have traditionally been the primary sensor for sewer inspections, while SSET and systems such as the

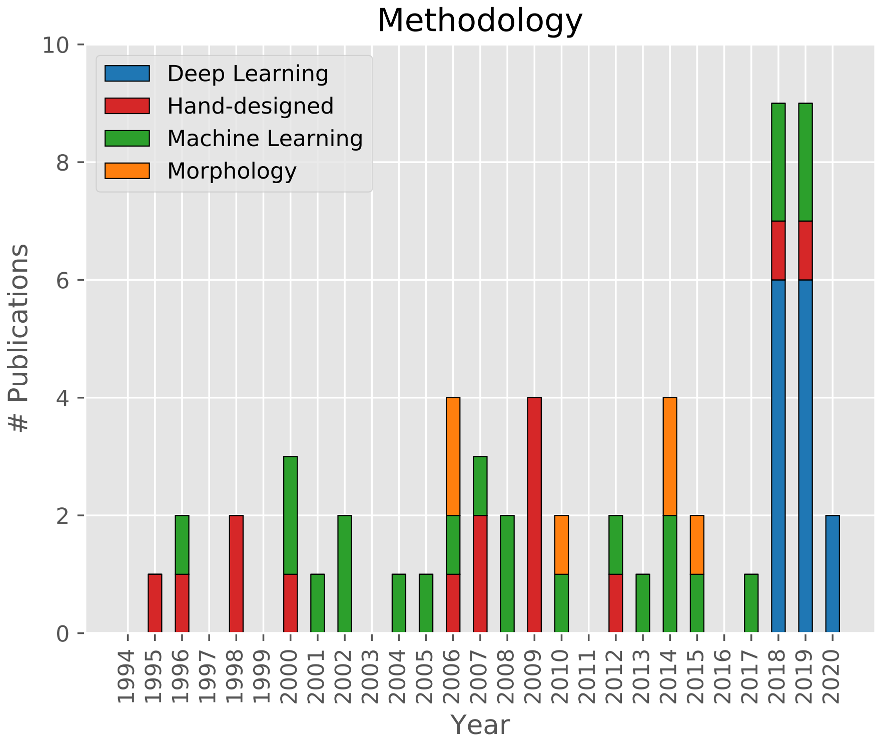
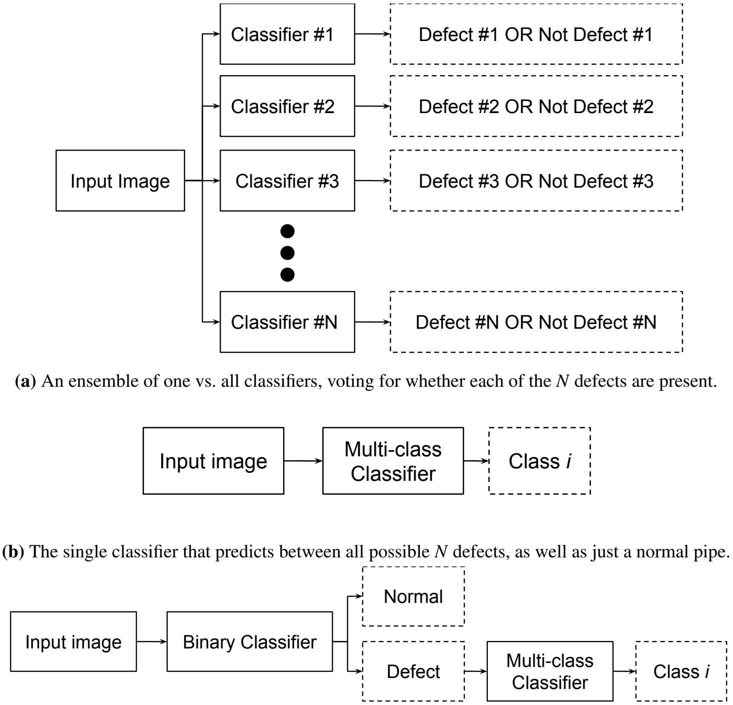


Fig. A.6: Distribution of the general pipeline methodologies used throughout the years. Based on the papers in Table A.1.

IBAK PANORAMO in general have been attempts at gaining a market foothold with potentially more advanced, but also more expensive, systems, which have not been widely adopted. While the IBAK PANORAMO is still utilized, research utilizing SSET images has stagnated, after prominent usage by some author groups in the early and mid-00's.

In the algorithmic stages there have been several trends throughout the years. In the mid 90's and early 00's, there were clear tendencies on utilizing pipelines based on simple edge based and geometric approaches of grayscale images, which were used over color images due to better contrast. The resulting augmentations were analyzed with heuristic rule-based decision systems in order to detect and segment defects. These methods were subsequently supplemented by the application of MLPs and other artificial neural network approaches based on hand-picked geometric and intensity features. Around the turn of the century, morphological approaches became increasingly popular for both segmentation and classification of regions of interests. This branch of segmentation and classification together with research into edge detectors designed specifically



(a) An ensemble of one vs. all classifiers, voting for whether each of the *N* defects are present.

(b) The single classifier that predicts between all possible *N* defects, as well as just a normal pipe.

(c) The two stage hierarchical pipeline

Fig. A.7: Illustration of the three different classification pipelines, shown for the multi-class classification problem when considering *N* defects. Dashed boxes shows output of the classifiers.

to detect cracks, led to a branch of research in the field, which is still active. However, these morphology and crack detector based approaches are hindered by the fact that the methods are designed for specific defects or environments, and therefore do not necessarily generalize well. There has therefore simultaneously been a push towards machine learning based approaches combined with semi hand-crafted and designed feature descriptions. These feature descriptions are often based on underlying patterns in the frequency or spatial domains of the entire image that may not be immediately obvious for humans. By analyzing a feature representation of the entire image, the problem at hand is generally shifted from designing hand-crafted pipelines for detection, segmentation, and classification, into choosing representative feature descriptors and machine learning algorithms. This branch of the field was first investigated in 2003 and has been widely applied since 2008, and as of 2018 the advances of deep learning and CNNs have been utilized. Simultaneously, color images are more utilized, as methods are now more capable of incorporating the extra information. Lastly, there have been very few efforts made with regard to making predictions temporally coherent. Only Myrans *et al.* [72] have evaluated how temporal filtering affects the consistency of the defect predictions of modern classification systems. The feature description and general pipeline methodologies over the years are shown in Figure A.5 - A.6, based on the methods in Table A.1.

Within the machine learning branch there has been three clear trends with regard to primarily the multi-class and multi-label classification tasks. The systems classify images using either an ensemble of binary classifiers, a two stage hierarchical approach consisting of a binary and multi-class detector, or a single multi-class/label classifier. These methodologies are illustrated in Figure A.7. Each of these methods have strengths and weaknesses which should be considered. An ensemble of binary classifiers breaks the complex problem at hand into several smaller and potentially easier problems, which may be solved using simpler models. This approach leads to a multi-label approach for free, as several labels can be assigned if each model outputs a prediction score above a set threshold [77, 135]. However, the ensemble approach also leads to a more expensive system at the inference stage. The two stage hierarchical approaches similarly divide the complex problem into its constituents, primarily for the purpose of speed at the inference stage. A smaller model filters out all non-defective images, and only run the more complex multi-class/label system when a defect is present. Lastly, using a single multi-class/label model, allows the model to draw knowledge from the different classes and develop interconnected hierarchical rules. This is, however, at the cost of needing more data and resulting in a potentially more complex decision surface. The ensemble and two stage methods have been widely used due to the lower complexity of the model and need for less labeled data. It has, however, also meant that primarily smaller CNN models have been utilized and the potential of the advances within deep learning have not been fully explored within the field.

Based on the current and previous trends in the field, we believe that future advances and breakthroughs will be achieved by the continued adoption of state-of-the-art computer vision techniques. Specifically, the continued adoption of deep learning based techniques is required. This is based on the proven success of deep learning techniques within the image classification, localization, detection and segmentation tasks in the computer vision field. It seems as if the future trend is to move away from hand-designed methods, and that the focus of the field is now on employing more advanced machine learning based approaches, as seen within the last few years [82, 89, 134, 136]. In order to retain transparency in the decision making process, it may be beneficial to incorporate research from the emerging field of explainable artificial intelligence [140]. Similarly, if the intention is to deploy the developed systems in real world scenarios, further investigation of the temporal consistency of the produced algorithms are of high priority and deserves more interest and focus. It may also be beneficial to depart from developing a system which tries to classify defects in all scenarios, but instead develop specialized subsystems focusing on classifying defects for specific pipe types and materials, similar to the ensemble approach of classifying different defects.

Table A.1: Recent academic studies in image-based automated sewer inspections. The group column indicates author group affiliation. Author groups are inserted based on the newest paper of the group. Corresponding datasets and evaluation protocols for each paper are available in Table A.2.

Paper	Year	Group	Acquisition	Pre-processing	Detection/Segmentation	Feature Descriptor	Classifier	Temporal Filtering
[83]	2020	1, 2	CCTV	-	-	-	CNN	-
[78]	2019	1	CCTV	-	-	-	CNN	-
[77]	2018	1	CCTV	-	-	-	CNN	-
[94]	2001	1	SSET	Grayscale Conv., Per-class Operations, Binarization	-	-	MLP	-
[82]	2019	2	CCTV	-	-	-	CNN CRF	-
[80]	2018	2	CCTV	-	-	-	CNN	-
[91]	2020	-	CCTV	-	-	-	CNN	-
[134]	2019	-	CCTV (Fish-eye)	-	-	-	CNN	-
[85]	2019	-	CCTV	-	-	-	CNN	-
[89]	2019	-	CCTV	-	-	-	CNN	-
[136]	2019	-	CCTV (Zoom)	-	-	-	CNN	-
[88]	2019	-	CCTV	Graysacle Conv.	-	Hu Invariant Moments, Image Statistics, Lateral Fourier Transform Statistics, Daubechies Wavelet Statistics	SVM	-
[90]	2019	-	CCTV	Grayscale Conv., Grayscale Morph., Histogram Stretching, Binarization	Floodfill, Binary Morph.	Height, Width	Rule-based	-
[123]	2019	-	CCTV (Unwrapped)	-	Sliding Window	LBP	OCSVM	-
[72, 74]	2018	3	CCTV	Grayscale Conv.	-	GIST	RF	HMM, OOF
[135]	2018	4	CCTV (Fish-eye)	Grayscale Conv.	Grayscale Morph., Gabor Filters, Active Contours, Distance Transform, Ellipse Fitting	Geometrical Attributes,	Rule-based	
[76]	2018	4	CCTV	Grayscale Conv., Gaussian Blur	-	Histogram of SIFT features	HMM, CNN	-

Continuation of Table A.1 ...

Paper	Year	Group	Acquisition	Pre-processing	Detection/Segmentation	Feature Descriptor	Classifier	Temporal Filtering
[75]	2017	4	CCTV	Grayscale Conv., Gaussian Blur	-	Spatio-temporal HOG	HMM	-
[122]	2018	-	CCTV (Unwrapped)	Image Enhancement, YCbCr Conv., Window Contrast Norm.	-	-	CNN	-
[87]	2018	-	CCTV	-	-	-	CNN	-
[86]	2018	-	CCTV	-	-	-	CNN	-
[114]	2015	-	SSET	Luminance Conv.	-	Contourlet, MR Filter Bank, GLCM Statistics	RotBoost	-
[65]	2015	-	CCTV	-	Sobel, Binary Morph., Skeletonization	Area	Rule-based	-
[58]	2014	5	CCTV	Grayscale Conv., Sobel, Hough	Binary Morph., Rule-based Linking	Area, Eccentricity	Rule-based	-
[59]	2014	5	CCTV	Grayscale Conv., Otsu, Median Blur	Binary Morph.	HOG	SVM	-
[120]	2014	6	CCTV (Unwrapped)	Median Blur	-	8×8 Windowed Haar Wavelet	SVM	-
[118]	2010	6	CCTV (Unwrapped)	HSB Conv.	SVM, Binary Morph., CCA	Shape, Position	Rule-based	-
[52]	2014	7	CCTV	Median Blur	Sobel, Binary Morph.	-	-	-
[47]	2008	7	CCTV	Grayscale Conv.	-	Discrete Wavelet Transform, GLCM Features	SVM	-
[63]	2013	-	CCTV	Grayscale Conv., Otsu	Binary Morph., Laplacian	Height, Width, Area	Neuro-Fuzzy	-
[116]	2012	8	CCTV (Unwrapped)	Grayscale Conv., Gaussian Blur, Canny	Hough, Graph Construction, Dijkstra	-	-	-
[115]	2007	8	CCTV (Unwrapped)	Grayscale Conv., Fourier Amp. Mod., FFT Noise Removal	Canny, Sliding Window	-	Multi-scale Analysis, Rule-based	-

Continuation of Table A.1 ...

Paper	Year	Group	Acquisition	Pre-processing	Detection/Segmentation	Feature Descriptor	Classifier	Temporal Filtering
[121]	2012	6, 8	CCTV (Unwrapped)	Gaussian Blur	Canny	RMS, Std. Dev.	SOM	-
[62]	2010	-	CCTV	Grayscale Conv.	K-means	Geometrical Attributes	Fuzzy Multi-factor Analysis	-
[131]	2009	9	CCTV (Fish-eye)	Image Registration, Histogram Equalization, Mean Blur	Image Diff.	-	Rule-based	-
[132]	2009	9	CCTV (Fish-eye)	Gaussian Blur, Canny	Region Growing	Color Histogram	Rule-based	-
[56]	2009	-	CCTV	HSV Conv., Grayscale Conv., Mean Blur, Histogram Equalization	QFCM, Binary Morph.	Height, Width	Rule-based	-
[55]	2009	-	CCTV	Grayscale Conv., Histogram Equalization, Contrast Stretching, Laplacian	Binary Morph.	Height, Width	Rule-based	-
[45]	2008	10	CCTV (Fish.eye)	-	-	-	CNN	-
[44]	2004	10	CCTV	-	-	-	CNN GA	-
[41]	2002	10	CCTV	Subdivision	-	Wavelet Packet Transform, PCA	MLP	-
[42]	2002	10	CCTV	Mean Blur, Subdivision	-	Haar Wavelet Transform, Shannon Entropy	Logistic Regression	-
[127]	2007	-	CCTV (Fish-eye)	Unwrapping, Brightness Conv., Gaussian Blur, Sobel	Sliding Window	Similarity	Rule-based	-
[54]	2007	-	CCTV	-	-	-	MLP	-
[113]	2006	11	SSET	Image Enhancement	Grayscale Morph., LoG	Area	Rule-based	-
[109, 110]	2006	11	SSET	LDA Grayscale Conv.	Grayscale Morph., Otsu, Image Diff.	Area	Rule-based	-

Continuation of Table A.1 ...

Paper	Year	Group	Acquisition	Pre-processing	Detection/Segmentation	Feature Descriptor	Classifier	Temporal Filtering
[107]	2006	11	SSET	Grayscale Conv.	Ratio Crack Detector, Cross-Correlation Crack Detector, Response Fusion, NN Linking	-	-	-
[108]	2006	11	SSET	-	-	Geometrical Attributes	Neuro-Fuzzy	-
[40]	2005	12	CCTV	-	Grayscale Morph., Background Sub., Thresholding	Geometrical/Intensity Attributes	MLP	-
[38]	2000	12	CCTV	-	-	Geometrical/Intensity Attributes	MLP	-
[36]	2000	-	CCTV	Binary Conv., Median Blur, Laplacian	Hough, Ellipse Fitting	-	-	-
[35]	2000	-	CCTV	-	Support Map	Vectorized Image, PCA	RBF, Posterior Map	Q-learning
[31]	1998	13	CCTV	Median Blur, Histgoram Equalization	Canny, Curvature Estimation	-	Rule-based	-
[30]	1996	13	CCTV	-	Canny, Reflective Photometric Stereo	-	Rule-based	-
[33]	1998	-	CCTV	Conv. Edge Det., Otsu	Line Based Thinning, Fitted Circle, Fourier Descriptors	-	Rule-based	-
[32]	1996	-	CCTV	Image Unwrapping, Contrast Enhancement	SBCS	-	SOM	-
[27]	1995	-	CCTV	Canny, Arc Grouping	Fitted Ellipse	-	Rule-based	Matching

5 Dataset and Evaluation

The evaluation of an algorithm consists of two main elements: the dataset and the evaluation metric. The dataset is of immense importance when testing an algorithm. If the dataset is not representative of the type of environment the algorithm should be used in or simply not large enough to represent the inherent variability of each defect class, it becomes impossible to state whether the algorithm can generalize. Equally important is the evaluation metric. If a poor metric is used, it is impossible to state whether an algorithm generalizes well.

We have investigated the characteristics of the datasets of the papers presented in Table A.1 and the utilized evaluation metrics in these papers. This is represented in Table A.2 and described in the following sections. The performance of each method is not reported due to difference in task, metrics, and datasets, meaning the methods are not directly comparable. We report information as it is explicitly written in the papers and do not intend on extrapolating hidden information. Furthermore, to make the data comparable, we try to report the top level information. This means that *e.g.* when predetermined regions are used for training, we only report the amount of regions of interest, if and only if, the amount of images used for the evaluation process is not stated; otherwise, we state the amount of images. We have investigated how the datasets are constructed, in respect to the algorithmic and environmental considerations, as well as the metrics used for evaluating the performance of the methods.

5.1 Algorithmic Considerations

The datasets are designed for widely different tasks. Some work with the image classification task either on a binary, multi-class, or multi-label level, which is increasingly harder. The classification task can be further generalized into the defect detection and the defect segmentation tasks. Defect detection consists of localizing and classifying areas with defects, while the defect segmentation task consists of assigning a label to each individual pixel in the image.

The segmentation tasks have been in focus for many years as part of crack detection and segmentation. On top of these systems, researchers have built systems to perform the multi-class and multi-label tasks such as classifying the type of cracks. In some cases, the terms anomaly detection/segmentation are used to indicate that the class is not specified but that a defect is detected or segmented [123, 127].

The evaluation protocol of the datasets has been investigated. The most common approach for evaluation is to utilize a predetermined train, validation, and test split. The validation set should be used as a test set while training in order to gauge the performance, whereas the test set should ideally only be evaluated once when the final model has been chosen based on the validation score. However, k-fold cross validation (CV) is often used when there is not enough data to make the representative data splits. In these cases, the average of the resulting metric for each fold is reported. This has been done in different ways. Kumar *et al.* [77, 83] used 5-fold CV to generate different train, validation, and test data splits while Meijer *et al.* [134] used a variation of a 30-fold CV dubbed *leave-two-inspections-out* where 2 out of 30 inspections in the dataset were excluded and saved for validation and testing. Halfawy and Hengmeechai [59] and Mashford *et al.* [120] used CV on the training split in order to determine the parameters of the used SVM. The traditional application of CV where the data is split into training and testing splits has also been utilized by several authors, with varying amount of folds [35, 42, 44, 74, 88, 114]. Unfortunately, several papers do not explicitly state what is in the dataset or how it is evaluated.

5.2 Environment Considerations

With all the previous algorithmic considerations, it is also important to consider the actual environments represented by the datasets. These considerations are, however, not always explicitly reported in the literature.

There are three main components when considering the investigated physical pipes: the material, the shape, and the size of the pipes. Likewise, the inspection guide used to label the defects and the country where the data is recorded is also of great interest, as this will effect what defines the different classes of defects, and potentially the rate of defects, respectively.

Rigid pipe materials have been investigated extensively, specifically: concrete [30, 31, 34–36, 38, 40, 55, 58, 59, 65, 72, 74, 75, 77, 78, 83, 88, 89, 107, 108, 110, 113, 116, 118, 120, 122, 123, 132, 134], clay [38, 40, 65, 91, 94] and vitrified clay [33, 52, 58, 59, 72, 74, 77, 78, 83, 115, 132, 135], brick [72, 74], iron [77], and stoneware [116, 122]. Flexible pipe materials have been less investigated, with primarily PVC pipes [78, 83, 88, 89] being used, with instances of HDPE [88, 89] and generic plastic pipes [116].

Similarly, datasets has primarily consisted of circular pipes [33–36, 58, 59, 72, 74, 75, 78, 89, 122, 123], with instance of rectangular [89], egg [72, 74], and horseshoe [72, 74] shapes also being included in the datasets. The size of the pipes has primarily been below 1000 mm diameter. In some cases, larger pipes with a diameter up to 3100 mm, known as “tunnels” have also been investigated.

Geographically, the investigated datasets originates from North America [58, 59, 77, 78, 80, 82, 83, 91, 94, 107, 108, 110, 113, 131, 132], China [88, 89], Taiwan [47, 52], Japan [41, 127], South Korea [85, 90], Australia [65, 120], the Netherlands [134], the United Kingdom [72, 74], Germany [34–36, 115], and Qatar [135]. Currently, four standards have been explicitly stated as being followed when labeling data: the Manual of Sewer Condition Classification (MSCC) from the British Water Research Center (WRc), the Pipeline Assessment Certification Program (PACP) from the American National Association of Sewer Service Companies (NASSCO), the European standard EN 13508-2, and an unnamed standard from the Japan Sewage Works Association (JSWA) followed by Ahrary *et al.* [127]. The PACP guide is an adaption of the MSCC guide made in collaboration with WRc.

5.3 Metrics

Results can be deceptive, without a good metric. There are, however, no set consensus on which metrics to use and how to report the results.

For the classification tasks, the most common metric used is accuracy [47, 72, 74, 76–78, 85, 86, 88, 94, 108, 114, 115, 135, 136]. Additionally, auxiliary binary metrics have also been utilized: precision [77, 86, 87, 89, 135, 136], recall [77, 86, 87, 89, 134–136], F1-score [86, 89, 136], and the confusion matrix [72, 74, 77, 85, 87, 89, 108, 114, 135]. These metrics are further summarized by using the Receiver Operator Curve (ROC) [72, 86, 87, 89, 134] and the Precision-Recall (PR) curve [134], together with the area under the curves, AUROC [72, 87, 89, 134] and AUPR [134]. Similarly, Meijer *et al.* proposed using conditional metrics when working with a large imbalanced dataset, namely investigating the specificity and precision at variable recall values. Myrans *et al.* [74] proposed reporting the accuracy when the ground truth was allowed to be within the *top-k* predictions, for $k = \{1, 2, 3, 4\}$, similar in style to how ImageNet [139] is evaluated. Ganegarada *et al.* manually inspected how the different features were clustered [121].

A different set of metric is used for the detection task. Cheng and Wang [80] measured performance based on the per-class Average Precision (AP), the mean Average Precision (mAP), and the per-class PR curves and missing rate curves. Kumar and Abraham [78] and Kumar *et al.* [83] similarly report the AP. Accuracy is reported by several authors [27, 35, 41, 42, 75, 131], as well as the True Positive Rate (TPR) [35, 127] and False Positive Rate (FPR) [127, 131], and the confusion matrix [75, 131]. Kirstein *et al.* [116] used the F1-score, Paletta and Rome [35] additionally report the amount of false positives, Moradi and Zayed [75] report recall and precision, and Guo *et al.* [131] also report the "true accuracy", which is equal to the accuracy metric but without false positives counted. Yin *et al.* [91] report the mAP, precision, recall, F1-score, and the confusion matrix. In several cases, no numerical evaluation was reported [30–33, 36].

Similar to the object detection task, the segmentation task is evaluated using a third set of metrics building upon the classification metrics. Accuracy [39, 40, 59, 120, 122, 132] and the confusion matrix [59, 110, 120, 123, 132] are reported alongside the recall [59, 123, 132] and precision [59, 123]. Künzel *et al.* [122] and

Wang and Cheng [82] report the mean Intersection over Union (mIoU) metric often used for instance and semantic segmentation tasks. Wang and Cheng also report the pixel accuracy per class, the mean pixel accuracy, and the frequency-weighted mIoU. Mashford *et al*. [120] report the Overall Success Rate (OSR), which is the percentage of true predictions in the whole image, Guo *et al*. [132] report the FPR, and Halfawy and Hengmeechai [59] report the ROC and AUROC. Swarnalatha *et al*., and Xue-Fei and Hua did not report any numerical evaluation [55, 56].

For the subfield of crack segmentation, accuracy [44, 45, 58, 62, 63], the confusion matrix [58, 107] or parts of it [90, 113], precision [54, 58], and recall [58] are still reported. Halfawy and Hengmeechai [58] also report the FPR. These metrics are, however, also supplemented by metrics trying to quantify aspects such as the quality, correctness, completeness, and redundancy of the segmentation [52, 107, 113]. In many cases, the evaluation is conducted using a "buffer" method where the detections and ground truth images are morphological dilated and compared in different ways [110, 113, 120].

5.4 Discussion and Future Directions

In the following, we will discuss the transparency, the datasets, and the used metrics of the investigated literature.

Transparency

From this survey, it is clear that very few papers openly share data and code implementation. Throughout all papers investigated in this survey, only Myrans *et al*. [72] and Xie *et al*. [136] state that data can be acquired by request, and only Xie *et al*. share their code in a public repository. Heo *et al*. [90] directly describe the used MATLAB functions, while Chae share the utilized MATLAB scripts in their doctoral dissertation [141]. Lastly, Su *et al*. [51–53] share the MATLAB code for the MSED algorithm directly in the papers.

This means that all researches utilize local dataset, design their algorithms towards the specific dataset statistics, and rarely compare with previous methods in order to determine whether the proposed algorithm improves on the state-of-the-art methods.

Datasets

Within recent years' data-driven methods have been favored, which has led to a focus on the classification tasks with few general defect detection and segmentation algorithms. Concurrently, it is clear that the usage of machine learning and data-driven algorithms has led to a surge in the amount of the data used. For many papers in the early to mid-00's, only a few hundred or thousand images were used. Comparatively, the recent

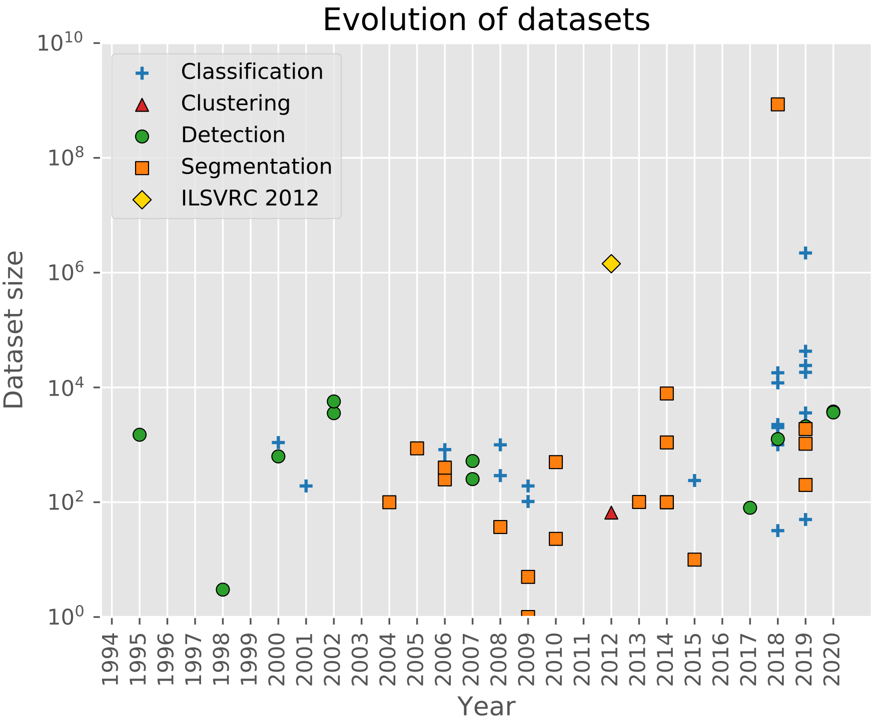


Fig. A.8: Evolution of the stated datasets used over the years for different algorithmic tasks. Evaluation protocols in Table A.2 with valid dataset sizes are shown. For comparison the dataset used in the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [142] is also shown. Note that the y-axis is log-scaled.

papers using CNNs have used between 18,000 and 2,200,000 images for the classification task [77, 78, 85–87, 89, 134, 136], and in the case of semantic segmentation, 8.5 billion pixels were labeled [122]. The datasets are further extended through the use of data augmentations, that might increase the dataset size by a factor of 1000 [77]. The increase in dataset sizes and focus on classification tasks is shown in Figure A.8.

Additionally, the size of dataset is connected with the complexity of the task. The complexity increases with the number of classes to detect and classify, necessitating a larger dataset in order to capture the inherent variance and key elements of the classes. This has, however, not been consistent as some researches have focused on classifying a single defect, such as cracks, while others try classifying 13 different classes [74]. Similarly, the ratio of defects and balance of class data has an equally important impact. If the dataset is imbalanced by containing a majority of images from a single class *e.g.* normal pipes the algorithm can be biased towards this class and ultimately be unusable for the classification task. There is, however, not a clear consensus in the literature on how to handle this. Meijer *et al.* [134] argue that since the vast majority of sewer stretches do not contain defects, this should be represented in the dataset. Their dataset consisting of more than 2.2 million images, therefore, only contain approximately 18,000 images with defects, which corresponds to a defect ratio of 0.8%. In order to counteract this imbalance when training, Meijer et al, Li *et al.* [89], and Tennakoon *et al.* [87] oversample the defective images while Yang and Su [47] compare sampling with equal and class based prior probabilities. Other attempts consists of grouping classes with few samples together in an "else" class [89], grouping images with several classes into a "multiple" class [74], and using data from other classes, which are not in the classes when evaluating [77, 136].

When looking at the contents of the datasets, there are clear tendencies towards rigid, specifically concrete, circular non-man-entry pipes. This makes sense from a historic perspective as these materials are commonly used. However, pipe materials such as PVC and HDPE have only been investigated in the literature as of 2012. There is thus a lack of research applied to inspection of flexible pipes that not only look different, but also have defects manifesting themselves in different ways compared to rigid pipes. Even more noticeable is the lack of datasets investigating pipes restored by having a lining applied. Similarly, the focus has been primarily on small circular pipes. While by far the prominent pipes shape and size used, small circular pipes not be the sole focus when considering pipe shapes and sizes. Information regarding the condition during the inspections such as whether the pipe was flushed beforehand or whether it is the main or lateral pipes is also often not stated.

There have also been tendencies in the literature to not explicitly state where the data is recorded or how it was labeled. When the inspection guide used for labeling defects is not explicitly stated, it becomes impossible to know what a classified defect truly encompasses and represents. Furthermore, based on the results presented by Dirksen *et al.* [3], the quality of the human annotations is not perfect, resulting in up to 25% false negatives, when using the EU standard. This is problematic, as it directly affects the quality of the data, and thereby the trained algorithms. A potential solution could be to have a group of inspectors each review all, or a subset, of the data, and determine the uncertainty for each defect class based on the variability of the inspectors annotations. This is, however, cumbersome and expensive, and therefore unrealistic for large amount of data. The problem may also be handled by considering algorithmic approaches, looking into research areas such as robust algorithm and noisy label training [143]. Similarly, assuming a large enough dataset is provided, where the majority of each class is correctly labeled, a probabilistic approach may be adopted in order to quantify the uncertainty of the classification.

There is currently no clear consensus on what datasets should consist of, how many images and the kind of images should be included, the kind of pipes investigated, or how the data is labeled. All of these makes it immensely difficult to actually compare any reported results, and determine whether the methods and the selected parameters actually generalize across different sewer networks.

Additionally, there are currently differing opinions on whether the datasets should be balanced according to the amount of classes in the dataset, or represent the actual distribution of defects. We believe that using datasets that are heavily skewed towards one class is not the ideal approach, even if it matches the real life distribution. The majority of data in the skewed dataset would be practically unusable, as it will potentially lead to overfitting for that single class, and not properly learn to solve the classification tasks. Similarly, the dataset should on the other hand not consists of a large amount of defects following a long tailed distribution, as some classes will simply not have enough samples to be properly learned by the currently employed methods. This is, however, problematic as the most severe defects are also often the rarest. We therefore believe that the dataset should contain a balanced amount of normal and defective images, where the defective images contain the different defect class of interests with a suitable amount of samples. In order to still classify the rarest and arguably most important defects, we suggest excluding these defects from the general classification task, and instead employ methodologies specifically designed to classify these rare and sparser cases.

Metrics

There is currently no definitive metric within the field of automated sewer inspection. The most common classification metric is accuracy, which is only a good metric on balanced datasets. In order to account for this, some authors have also used precision, recall, the F1-score, and the confusion matrix. These are, however, by no means consistently reported. Furthermore, these are binary metrics that need to be averaged in some way across all classes. This is accounted for by reporting the ROC and PR curves, which are further summarized using the AUROC and AUPR. The ROC curve is, however, similarly to accuracy, not a good metric when working with imbalanced datasets [144].

The division in used metrics becomes even more apparent for the detection and segmentation tasks. There are currently no clear tendencies in which metric to use. The closest to a common metric is the accuracy metric, which is as earlier mentioned not a reliable metric for imbalanced datasets.

Based on the conducted survey, we are proposing the following set of recommendations that we believe will streamline the research conducted within this field, and in general improve the field as a whole.

1. We believe that in order to properly foster transparent and fair research, public and freely available datasets are needed as benchmark tasks. When looking at

the impact that the ImageNet [139] and the COCO [145] datasets and challenges have had within the computer vision field as a whole, it is clear that many of the advances from the last 10 years would not have been possible otherwise. Furthermore, public and free datasets will lead to a lower bar of entry into the field and easier collaboration on research ideas, which is expected to further improve research within this field.

2. A standardized set of metrics has to be adopted within the field. For the detection and segmentation tasks, the mAP and mIoU metrics used by the COCO challenges are obvious choices. For the classification tasks, it is less clear cut, other than it is clear that the commonly used accuracy metric should not be used for imbalanced datasets. We believe that averaged precision, recall, and F1-scores are good choices as auxiliary metrics, in conjunction with the confusion matrix and PR curve. As for the main metric, a top-k accuracy metric as calculated in the ImageNet classification challenge may be fitting.

3. We strongly believe that in order to make this research field more transparent, fair, and reproducible, it is important that the code used for publications is publicly available. This is common practice in several other computer vision fields by both industry and academic researchers.

Table A.2: Dataset of the academic efforts shown in Table A.1. The group column indicates author group affiliation. Author groups are inserted based on the newest paper of the group. Only information on still images are reported. All values reported are before processing or augmenting. The task column denotes how the method was evaluated. The defect column designates how large a percentage of the dataset which contain images with defects. It should be noted that the way Cross Validation (CV) is utilized is not consistent in the literature.

Paper	Year	Group	Task	Size (Frames)	Defect [%]	Classes	Data Split (trn/val/tst)[%]	CV Folds	Resolution (Pixels)	Pipe Material	Pipe Shape	Pipe Size [mm]	Inspection Guide	Country
[83]	2020	1,2	Defect Det.	3800	-	2	80/10/10	5	720 × 576 - 1507 × 720	Concrete, PVC, Vitrified Clay	-	203, 254, 305	-	USA
[78]	2019	1	Multi-class, Fracture Det.	3600, 2100	75, -	4, -	66.7/0/33.3, 85.7/0/14.3	-	-	Concrete, PVC, Vitrified Clay	Circular	203, 254, 305	-	USA
[77]	2018	1	Multi-label	12000	91.7	3	62.5/20.8/16.7	5	320×256 - 1440×720	Concrete, Iron, Vitrified Clay	Circular	203, 254	PACP	USA
[94]	2001	1	Multi-label, Cond. Asses.	192	-	4	10.4/0/89.6	-	638×1000	Clay	-	-	-	USA
[82]	2019	2	Defect Seg.	1885	-	4	64/16/20	-	512×256	-	-	-	-	USA
[80]	2018	2	Defect Det.	1260	-	4	75/10/15	-	320×256 - 1440×720	-	-	-	-	USA
[91]	2020	-	Defect Det.	3664	-	7	75/10/15	-	-	Clay	-	-	PACP	Canada
[134]	2019	-	Multi-label	2202582	0.8	12	-	30	1040×1040	Concrete	-	300-1000	EN 13508-2	NL
[85]	2019	-	Multi-class	24137	100	6	73.1/24.4/3.0	-	-	-	-	-	-	South Korea
[89]	2019	-	Multi-class	18333	46.1	8	70/15/15	-	296×166 - 1435×1054	Concrete, HDPE, PVC	Circular, Rectangular	300-2500	PACP	China
[136]	2019	-	Multi-class	42800	53.3	7	85/5/10, 80/8/12	-	-	-	-	-	-	-
[88]	2019	-	Multi-class	1045	100	7	-	-	-	Concrete, HDPE, PVC	-	225-1800	-	China
[90]	2019	-	Crack Seg.	200	-	-	-	-	240×320	-	-	-	-	South Korea
[123]	2019	-	Anomaly Seg.	50 (Mosaicks)	-	-	-	-	2592×1944	Concrete	Circular	200, 2325-3100	-	-
[74]	2018	3	Multi-class	2260	100	13	-	25	512×512	Brick, PVC, Vitrified Clay	Circular, Egg, Horseshoe	150-1500	MSCC	UK

Continuation of Table A.2 ...

Paper	Year	Group	Task	Size (Frames)	Defect [%]	Classes	Data Split (trn/val/tst)[%]	CV Folds	Resolution (Pixels)	Pipe Material	Pipe Shape	Pipe Size [mm]	Inspection Guide	Country
[72]	2018	3	Binary	1000	37.7	-	-	25	512×512	Brick, Concrete, Vitrified Clay	Circular, Egg, Horseshoe	200-1800	MSCC	UK
[135]	2018	4	Multi-label	32	-	4	-	-	-	Vitrified Clay	-	200	PACP	Qatar
[76]	2018	4	Crack Det.	-	-	-	-	-	640 × 480	-	-	-	-	Canada
[75]	2017	4	Anomaly Det.	80	50	-	0/0/100	-	640 × 480	Concrete	Circular	610	-	Canada
[122]	2018	-	Defect Seg.	8548431600 (Pixels)	5.68	9	80/0/20	-	600×1200	Concrete, Stoneware	Circular	200-500	-	-
[87]	2018	-	Multi-class	2000	100	5	0/0/100	-	720 × 576	-	-	-	-	-
[86]	2018	-	Multi-class	18000	44.4	5	75/0/25	-	-	-	-	-	-	-
[114]	2015	-	Multi-class	239	76.6	4	-	4	-	-	-	-	-	-
[65]	2015	-	Crack Seg.	10	-	-	-	-	-	Clay, Concrete	-	-	-	Australia
[58]	2014	5	Crack Seg.	100	50	-	0/0/100	-	320 × 240 352 × 240	Concrete, Vitrified Clay	Circular	250-900	-	Canada
[59]	2014	5	Root Seg.	1100	50	-	90.9/0/9.1	5	320 × 240	Concrete, Vitrified Clay	Circular	250-900	-	Canada
[120]	2014	6	Edge Seg.	7844 (Pixels)	-	-	80.9/0/19.1	N/A	-	Concrete	-	-	-	Australia
[118]	2010	6	Defect Seg.	23	-	3	-	-	-	Concrete	-	-	-	-
[52]	2014	7	Defect Seg.	100	100	2	20/0/80	-	-	Vitrified Clay	-	-	-	Taiwan
[47]	2008	7	Multi-class	291	100	4	13.7/0/96.3	-	-	-	-	-	MSCC	Taiwan
[63]	2013	-	Crack Seg.	101	-	-	-	-	-	-	-	-	-	-
[116]	2012	8	Flowline Det.	-	-	-	-	-	-	Concrete, Plastic, Stoneware	-	-	-	-
[115]	2007	8	Defect Det.	522	-	2	-	-	-	Vitrified Clay	-	-	-	Germany
[121]	2012	6, 8	Clustering	66	-	4	45.5/0/54.5	-	500×3840 - 500×50400	-	-	-	-	-
[62]	2010	-	Crack Seg.	500	-	-	-	-	-	-	-	-	-	-
[131, 132]	2009	9	Binary	103	49.5	-	0/0/100	-	-	Vitrified Clay	-	305	PACP	USA
[132]	2009	9	Binary	192	22.9	-	0/0/100	-	-	Concrete	-	-	PACP	USA
[56]	2009	-	Defect Seg.	1	-	2	-	-	-	-	-	-	-	-
[55]	2009	-	Defect Seg.	5	-	2	-	-	-	Concrete	-	-	-	-
[45]	2008	10	Crack Seg., Multi-class	37, 1000	-	-, 3	54/0/46, 16/0/84	-	320 × 240, -	-	-	-	-	-

Continuation of Table A.2 ...

Paper	Year	Group	Task	Size (Frames)	Defect [%]	Classes	Data Split (trn/val/tst)[%]	CV Folds	Resolution (Pixels)	Pipe Material	Pipe Shape	Pipe Size [mm]	Inspection Guide	Country
[44]	2004	10	Crack Seg.	100	-	-	-	25	20 × 20 - 320 × 240	-	-	-	-	-
[41]	2002	10	Crack Det.	5700 (Sections)	-	-	0/0/100	-	320 × 240	-	-	-	-	Japan
[42]	2002	10	Crack Det.	3560 (Sections)	52.9	-	90/0/10	10	320 × 240	-	-	-	-	-
[127]	2007	-	Anomaly Det.	253	-	-	-	-	640 × 480	-	-	-	JSWA	Japan
[54]	2007	-	Crack Seg.	-	-	-	-	-	-	-	-	-	-	-
[113]	2006	11	Crack Seg.	400	-	-	37.5/0/62.5	-	285×185, 900×550	Clay, Concrete	-	-	-	USA, Canada
[109, 110]	2006	11	Crack Seg.	300	-	5	60.6/0/39.4	-	-	Concrete	-	457	-	USA, Canada
[107]	2006	11	Crack Seg.	250	-	5	-	-	-	Concrete	-	457	-	USA, Canada
[108]	2006	11	Multi-class	500	68	14	60/0/40	-	-	Concrete	-	457	-	USA, Canada
[40]	2005	12	Infiltration Seg.	868 (Segments)	5.5	-	60/20/20	-	-	Clay, Concrete	-	200-1000	-	Canada
[38]	2000	12	Multi-class	1096 (Segments)	100	4	60/20/20	-	512 × 512	Clay, Concrete	-	-	-	Canada
[36]	2000	-	Lateral Det.	-	-	-	-	-	-	Concrete	Circular	600	-	Germany
[34, 35]	2000	-	Inlet Det.	111, 517	79.3, -	-	0/0/100, -	-, 6	-	Concrete	Circular	600	-	Germany
[30, 31]	1998	13	Lateral Det.	-	-	-	-	-	-	Concrete	-	300, 450	-	-
[33]	1998	-	Defect Det.	3	-	2	-	-	1024×512	Vitrified Clay	Circular	-	-	-
[32]	1996	-	Socket Det.	-	-	-	-	-	376 × 288	-	-	-	-	-
[27]	1995	-	Defect Det.	1498	-	2	-	-	768×576	-	-	-	-	-

6 Conclusion

This survey paper has provided a thorough overview of research within image-based automated CCTV and SSET sewer inspection over the last 25 years. The automated inspection pipeline was split into acquisition, pre-processing, detection and segmentation, feature description, classification, and temporal filtering, which were each investigated in depth. From this dive into the literature several trends throughout the years were uncovered showing the rise and fall of geometric and hand-crafted algorithm approaches, the increase in morphological based methods, and the current trend of utilizing more designed features and the usage of data-driven machine learning algorithms, specifically CNNs. Similarly, the utilized datasets and evaluation protocols were investigated in-depth, revealing tendencies to focus on rigid circular pipes, the usage of widely differently constructed datasets, and the lack of standardized evaluation metrics. Based on the conducted survey, we recommend that one or more free and publicly available datasets are created, with pre-defined standardized metrics, which will act as benchmarks for the research community. We also recommend the adoption of open-source code for each publication, in order to promote an open and reproducible research environment.

Acknowledgments

The authors would like to thank our colleagues Chris Holmberg Bahnsen and Marike Koch van den Broek for their invaluable discussions and feedback on iterations of the manuscript. This work is supported by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project. The authors declare no conflict of interests.

A.A Surveyed Papers

Table A.3: Overview of all papers which have been surveyed, listed chronologically. The author groups match those of Table A.1-A.2, with the addition of author groups 14-19.

Paper	Year	Group	Title
[83]	2020	1, 2	Deep Learning-based Automated Detection of Sewer Defects in CCTV Videos
[91]	2020	-	A Deep Learning-based Framework for an Automated Defect Detection System for Sewer Pipes
[78]	2019	1	A Deep Learning based Automated Structural Defect Detection System for Sewer Pipelines
[81]	2019	2	Semantic Segmentation of Sewer Pipe Defects using Deep Dilated Convolutional Neural Network
[82]	2019	2	A Unified Convolutional Neural Network Integrated with Conditional Random Field for Pipe Defect Segmentation
[85]	2019	14	Underground Sewer Pipe Condition Assessment based on Convolutional Neural Networks
[88]	2019	-	Diagnosis of Sewer Pipe Defects on Image Recognition of Multi-Features and Support Vector Machine in a Southern Chinese city
[89]	2019	-	Sewer Damage Detection from Recognition CCTV Inspection Data using Deep Convolutional Neural Networks with Hierarchical Classification
[90]	2019	-	Crack Automatic Detection of CCTV Video of Sewer Inspection with Low Resolution
[123]	2019	-	A Vision-based System for Internal Pipeline Inspection
[134]	2019	-	A Defect Classification Methodology for Sewer Image Sets with Convolutional Neural Networks
[136]	2019	-	Automatic Detection and Classification of Sewer Defects via Hierarchical Deep Learning
[70]	2018	3	Automatic Identification of Sewer Fault Types using CCTV Footage
[71]	2018	3	Using Automatic Anomaly Detection to Identify Faults in Sewers
[72]	2018	3	Automated Detection of Faults in Sewers using CCTV Image sequences
[73]	2018	3	Combining Classifiers to Detect Faults in Wastewater Networks
[74]	2018	3	Automated Detection of Fault Types in CCTV Sewer Surveys

Continuation of Table A.3 ...

Paper	Year	Group	Title
[76]	2018	4	Automated Sewer Pipeline Inspection using Computer Vision Techniques
[77]	2018	1	Automated Defect Classification in Sewer Closed Circuit Television Inspections using Deep Convolutional Neural Networks
[79]	2018	2	Development and Improvement of Deep Learning based Automated Defect Detection for Sewer Pipe Inspection using Faster R-CNN
[80]	2018	2	Automated Detection of Sewer Pipe Defects in Closed-Circuit Television Images using Deep Learning Techniques
[84]	2018	14	Utilizing Text Recognition for the Defects Extraction in Sewers CCTV Inspection Videos
[86]	2018	-	An Intelligent Sewer Defect Detection Method based on Convolutional Neural Network
[87]	2018	-	Visual Inspection of Storm-Water Pipe Systems using Deep Convolutional Neural Networks
[122]	2018	-	Automatic Analysis of Sewer Pipes based on Unrolled Monocular Fisheye Images
[135]	2018	4	Automated Defect Detection Tool for Closed Circuit Television (CCTV) Inspected Sewer Pipelines
[69]	2017	3	Automatic Detection of Sewer Faults using Continuous CCTV Footage
[75]	2017	4	Real-Time Defect Detection in Sewer Closed Circuit Television Inspection Videos
[66]	2016	15	3D Anomaly Inspection System for Sewer Pipes using Stereo Vision and Novel Image Processing
[67]	2016	3	Using Support Vector Machines to Identify Faults in Sewer Pipes from CCTV Surveys
[68]	2016	3	Automated Detection of Faults in Wastewater Pipes from CCTV Footage by using Random Forests
[53]	2015	7	Segmentation of Crack and Open Joint in Sewer Pipelines Based on CCTV Inspection Images
[61]	2015	5	Integrated Vision-based System for Automated Defect Detection in Sewer Closed Circuit Television Inspection Videos
[65]	2015	15	Dou-Edge Evaluation Algorithm for Automatic Thin Crack Detection in Pipelines

Continuation of Table A.3 ...

Paper	Year	Group	Title
[114]	2015	-	Classification of Defects with Ensemble Methods in the Automated Visual Inspection of Sewer Pipes
[52]	2014	7	Application of Morphological Segmentation to Leaking Defect Detection in Sewer Pipelines
[58]	2014	5	Efficient Algorithm for Crack Detection in Sewer Images from Closed-Circuit Television Inspections
[59]	2014	5	Automated Defect Detection in Sewer Closed Circuit Television Images using Histograms of Oriented Gradients and Support Vector Machine
[60]	2014	5	Optical Flow Techniques for Estimation of Camera Motion Parameters in Sewer Closed Circuit Television Inspection Videos
[64]	2014	16	A New Image Model for Predicting Cracks in Sewer Pipes based on Time
[120]	2014	6	Edge Detection in Pipe Images using Classification of Haar Wavelet Transforms
[63]	2013	16	A New Image-based Model For Predicting Cracks In Sewer Pipes
[116]	2012	8	Robust Adaptive Flow Line Detection in Sewer Pipes
[121]	2012	6, 8	Self Organising Map based Region of Interest Labelling for Automated Defect Identification in Large Sewer Pipe Image Collections
[50]	2011	7	Feature Extraction Of Sewer Pipe Defects using Wavelet Transform and Co-Occurrence Matrix
[51]	2011	7	Morphological Segmentation based on Edge Detection for Sewer Pipe Defects on CCTV Images
[119]	2011	6	Processing by SVM of Haar Wavelet Transforms for Discontinuity Detection
[21]	2010	7	No-Dig Inspection Technologies for Underground Pipelines
[49]	2010	7	Sewerage Rehabilitation Planning
[62]	2010	-	An Intelligent Fuzzy Multifactor based Decision Support System for Crack Detection of Underground Sewer Pipelines
[118]	2010	6	A Morphological Approach to Pipe Image Interpretation based on Segmentation by Support Vector Machine

Continuation of Table A.3 ...

Paper	Year	Group	Title
[48]	2009	7	Segmenting Ideal Morphologies of Sewer Pipe Defects on CCTV Images for Automated Diagnosis
[55]	2009	-	Automated Assessment Tool for the Depth of Pipe Deterioration
[56]	2009	-	Automated Assessment of Buried Pipeline Defects by Image Processing
[57]	2009	5	Video Processing Techniques for Assisted CCTV Inspection and Condition Rating of Sewers
[131]	2009	9	Automated Defect Detection for Sewer Pipeline Inspection and Condition Assessment
[132]	2009	9	Visual Pattern Recognition Supporting Defect Reporting and Condition Assessment of Wastewater Collection Systems
[133]	2009	9	Automated Defect Detection in Urban Wastewater Pipes using Invariant Features Found in Video Images
[45]	2008	10	Convolutional Neural Networks for Image Processing with Applications in Mobile Robotics
[46]	2008	7	Feature Extraction of Sewer Pipe Failures by Wavelet Transform and Co-Occurrence Matrix
[47]	2008	7	Automated Diagnosis of Sewer Pipe Defects based on Machine Learning Approaches
[130]	2008	9	Imagery Enhancement and Interpretation for Remote Visual Inspection of Aging Civil Infrastructure
[54]	2007	-	Decision-Support System for the Rehabilitation of Deteriorating Sewers
[115]	2007	8	Objective Condition Assessment of Sewer Systems
[117]	2007	6	Pixel-based Colour Image Segmentation using Support Vector Machine for Automatic Pipe Inspection
[126]	2007	17	Experimental Evaluation of Intelligent Fault Detection System for Inspection of Sewer Pipes
[127]	2007	17	An Automated Intelligent Fault Detection System for Inspection of Sewer Pipes
[129]	2007	9	Automatic Defect Detection and Recognition for Asset Condition Assessment: A Case Study on Sewer Pipeline Infrastructure System
[107]	2006	11	Automated Detection of Cracks in Buried Concrete Pipe Images
[108]	2006	11	Neuro-Fuzzy Network for the Classification of buried Pipe Defects

Continuation of Table A.3 ...

Paper	Year	Group	Title
[109]	2006	11	Segmentation of Buried Concrete Pipe Images
[110]	2006	11	Morphological Segmentation and Classification of Underground Pipe Images
[113]	2006	11	Segmentation of Pipe Images for Crack Detection in Buried Sewers
[125]	2006	17	Detecting Pipe Feature Points for Sewer Pipe System based on Image Information
[128]	2006	9	Automatic Visual Data Interpretation for Pipeline Infrastructure Assessment
[40]	2005	12	Automated Detection and Classification of Infiltration in Sewer Pipes
[111]	2005	11	Automated Condition Assessment of Buried Sewer Pipes based on Digital Imaging Techniques
[112]	2005	11	A Robust Approach for Automatic Detection and Segmentation of Cracks in Underground Pipeline Images
[44]	2004	10	Genetic Algorithm Optimization of a Convolutional Neural Network for Autonomous Crack Detection
[106]	2004	11	Intelligent System for Condition Monitoring of Underground Pipelines
[16]	2003	1	Computerized Sewer Pipe Condition Assessment
[43]	2003	10	Convolutional Neural Networks for Image processing: An Application in Robot Vision
[105]	2003	11	Computer Vision Techniques for Automatic Structural Assessment of Underground Pipes
[124]	2003	10	Convolutional Neural Networks for Robot Vision: Numerical Studies and Implementation on a Sewer Robot
[39]	2002	12	Automated Inspection of Utility Pipes: A Solution Strategy for Data Management
[41]	2002	10	Visual Feature Extraction via PCA-based Parameterization of Wavelet Density Functions
[42]	2002	10	Wavelet Entropy-based Feature Extraction for Crack Detection in Sewer Pipes
[104]	2002	11	Classification of Underground Pipe Scanned Images using Feature Extraction and Neuro-Fuzzy Algorithm
[94]	2001	1	Neuro-Fuzzy Approaches for Sanitary Sewer Pipeline Condition Assessment
[102]	2001	11	Automated Condition Assessment of Buried Sewer Pipeline using Computer Vision Techniques
[103]	2001	11	Development of an Automated Pipeline Inspection System

Continuation of Table A.3 ...

Paper	Year	Group	Title
[35]	2000	18	Learning Fusion Strategies for Visual Object Detection
[36]	2000	-	3D Interpretation of Sewer Circular Structures
[38]	2000	12	Classification of Defects in Sewer Pipes using Neural Networks
[92]	2000	1	Utilizing Neural Networks for Condition Assessment of Sanitary Sewer Infrastructure
[93]	2000	1	Automated Condition Assessment of Sanitary Sewer Pipelines
[100]	2000	11	Computer Vision Techniques for Inspection of Pipes
[101]	2000	11	Automated Segmentation of Underground Pipe scanned Images
[34]	1999	18	Visual Object Detection for Autonomous Sewer Robots
[37]	1999	12	Automated Detection of Surface Defects in Water and Sewer Pipes
[95]	1999	11	Underground Pipe Cracks Classification using Image Analysis and Neuro-Fuzzy Algorithm
[96]	1999	11	Automated Underground Sewage Pipe Condition Assessment by Image Analysis of the State-of-the-Art Sewer Scanner and Evaluation Technology Surveys
[97]	1999	11	Novel System for Automatic Underground Pipe Distress Detection and Classification
[98]	1999	11	Condition Assessment of Underground Sewer Pipes using a Modified Digital Image Processing Paradigm
[99]	1999	11	Automated Analysis and Detection of Cracks in Underground scanned Pipes
[31]	1998	13	Automatic Visual Detection of Lateral Junctions in Sewers
[33]	1998	-	Sewer Pipe Deformation Assessment by Image Analysis of Video Surveys
[30]	1996	13	Automatic Task Modelling for Sewer Studies
[32]	1996	-	Sewage Pipe Image Segmentation using a Neural based Architecture
[27]	1995	19	Robust Tracking of Circular Features
[29]	1995	13	Three-dimensional Description of Sewer Laterals via Reflective Photometric Stereo
[26]	1994	19	Detection and Tracking of Pipe Joints in Noisy Images
[28]	1994	13	Sensing for Feature Identification in Sewers

References

[1] American Society of Civil Engineers, “2017 infrastructure report card - wastewater,” 2017, accessed: 06-09-2019. [Online]. Available: https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf

[2] D. Huang, X. Liu, S. Jiang, H. Wang, J. Wang, and Y. Zhang, “Current state and future perspectives of sewer networks in urban china,” *Frontiers of Environmental Science & Engineering*, vol. 12, no. 3, p. 2, Feb 2018.

[3] J. Dirksen, F. H. Clemens, H. Korving, F. Cherqui, P. L. Gauffre, T. Ertl, H. Plihal, K. Müller, and C. T. Snaterse, “The consistency of visual sewer inspection data,” *Structure and Infrastructure Engineering*, vol. 9, no. 3, pp. 214–228, 2013.

[4] A. J. van der Steen, J. Dirksen, and F. H. Clemens, “Visual sewer inspection: detail of coding system versus data quality?” *Structure and Infrastructure Engineering*, vol. 10, no. 11, pp. 1385–1393, 2014.

[5] R. Wirahadikusumah, D. M. Abraham, T. Iseley, and R. K. Prasanth, “Assessment technologies for sewer system rehabilitation,” *Automation in Construction*, vol. 7, no. 4, pp. 259 – 270, 1998.

[6] J. M. Makar, “Diagnostic techniques for sewer systems,” *Journal of Infrastructure Systems*, vol. 5, no. 2, pp. 69–78, 1999.

[7] O. Duran, K. Althoefer, and L. D. Seneviratne, “State of the art in sensor technologies for sewer inspection,” *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, April 2002.

[8] S. Costello, D. Chapman, C. Rogers, and N. Metje, “Underground asset location and condition assessment technologies,” *Tunnelling and Underground Space Technology*, vol. 22, no. 5, pp. 524 – 542, 2007, trenchless Technology.

[9] T. Hao, C. Rogers, N. Metje, D. Chapman, J. Muggleton, K. Foo, P. Wang, S. Pennock, P. Atkins, S. Swingler, J. Parker, S. Costello, M. Burrow, J. Anspach, R. Armitage, A. Cohn, K. Goddard, P. Lewin, G. Orlando, M. Redfern, A. Royal, and A. Saul, “Condition assessment of the buried utility service infrastructure,” *Tunnelling and Underground Space Technology*, vol. 28, pp. 331 – 344, 2012.

[10] Z. Liu and Y. Kleiner, “State of the art review of inspection technologies for condition assessment of water pipes,” *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

[11] J. M. Mirats Tur and W. Garthwaite, “Robotic devices for water main in-pipe inspection: A survey,” *Journal of Field Robotics*, vol. 27, no. 4, pp. 491–508, 2010.

[12] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, “A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure,” *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196 – 210, 2015, infrastructure Computer Vision.

[13] S. Moradi, T. Zayed, and F. Golkhoo, “Review on computer aided sewer pipeline defect detection and condition assessment,” *Infrastructures*, vol. 4, no. 1, p. 10, Mar 2019.

[14] L. Corkill and D. Bennett, “Cctv in the united states,” 2006, accessed: 10-09-2019. [Online]. Available: http://www.sewerhistory.org/articles/maint/cctv_US_History.pdf

[15] IBAK Helmut Hunger GmbH & Co. KG, “Panoramo system,” accessed: 02-09-2019. [Online]. Available: https://www.ibak.de/en/produkte/ibak_show/frontenddetail/product/panoramo-system/

[16] M. J. Chae, T. Iseley, and D. M. Abraham, “Computerized sewer pipe condition assessment,” in *New Pipeline Technologies, Security, and Safety*, 2003, pp. 477–493.

[17] E. Rome, J. Hertzberg, F. Kirchner, U. Licht, and T. Christaller, “Towards autonomous sewer robots: the makro project,” *Urban Water*, vol. 1, no. 1, pp. 57 – 70, 1999.

[18] R. Kirkham, P. D. Kearney, K. J. Rogers, and J. Mashford, “Pirat—a system for quantitative sewer pipe assessment,” *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 1033–1053, 2000.

[19] A. Ahrary, A. A. Nassiraei, and M. Ishikawa, “A study of an autonomous mobile robot for a sewer inspection system,” *Artificial Life and Robotics*, vol. 11, no. 1, pp. 23–27, Jan 2007.

[20] D. Alejo, C. Marques, F. Caballero, P. Alvito, and L. Merino, “Siar: An autonomous ground robot for sewer inspection,” in *Proceedings of the Spanish Actas de las Jornadas de Automátic*, 2016, pp. 1–8, URL: https://robotics.upo.es/papers/ja_siar_final.pdf.

[21] M.-D. Yang, T.-C. Su, T.-Y. Wu, and K.-S. Huang, “No-dig inspection technologies for underground pipelines,” *Journal of GeoEngineering*, vol. 5, no. 3, pp. 99–104, Dec 2010.

[22] M.-D. Yang, T.-C. Su, N.-F. Pan, and Y.-F. Yang, “Systematic image quality assessment for sewer inspection,” *Expert Systems with Applications*, vol. 38, no. 3, pp. 1766 – 1776, 2011.

[23] S. Fu, D. Cooper, T. Pridmore, and N. Taylor, “Determination of sewer structure using a moving camera,” in *12th International Symposium on Automation and Robotics in Construction (ISARC), Warsaw, Poland*, 1995, pp. 547–554.

[24] T. Pridmore, D. Cooper, and N. Taylor, “Estimating camera orientation from vanishing point location during sewer surveys,” *Automation in Construction*, vol. 5, no. 5, pp. 407 – 419, 1997.

[25] D. Cooper, T. Pridmore, and N. Taylor, “Towards the recovery of extrinsic camera parameters from video records of sewer surveys,” *Machine Vision and Applications*, vol. 11, no. 2, pp. 53–63, Oct 1998.

[26] X. Pan, T. A. Clarke, and T. J. Ellis, “Detection and tracking of pipe joints in noisy images,” in *Videometrics 3*, vol. 2350, 1994, pp. 136–137.

[27] X. Pan, T. Ellis, and T. Clarke, “Robust tracking of circular features.” in *Proceedings of the British Machine Vision Conference*. BMVA Press, 1995, pp. 55.1–55.10.

[28] S. Broadhurst, T. Pridmore, and N. Taylor, “Sensing for feature identification in sewers,” in *11th International Symposium on Automation and Robotics in Construction (ISARC), Brighton, United Kingdom*, 1994, pp. 675 – 682.

[29] S. J. Broadhurst, T. P. Pridmore, N. Taylor, and G. Cockerham, “Three-dimensional description of sewer laterals via reflective photometric stereo,” in *12th International Symposium on Automation and Robotics in Construction (ISARC), Warsaw, Poland*, 1995, pp. 539–546.

[30] S. Broadhurst, G. Cockerham, N. Taylor, and T. Pridmore, “Automatic task modelling for sewer studies,” *Automation in Construction*, vol. 5, no. 1, pp. 61 – 71, 1996, 12th ISARC.

[31] N. Taylor, T. Pridmore, and S. Fu, “Automatic visual detection of lateral junctions in sewers.” *Proceedings of the Institution of Civil Engineers - Water, Maritime and Energy*, vol. 130, no. 2, pp. 56–69, 1998.

[32] J. R. del Solar and M. Köppen, “Sewage pipe image segmentation using a neural based architecture,” *Pattern Recognition Letters*, vol. 17, no. 4, pp. 363 – 368, 1996, neural Networks for Computer Vision Applications.

[33] K. Xu, A. Luxmoore, and T. Davies, “Sewer pipe deformation assessment by image analysis of video surveys,” *Pattern Recognition*, vol. 31, no. 2, pp. 169 – 180, 1998.

[34] L. Paletta, E. Rome, and A. Pinz, “Visual object detection for autonomous sewer robots,” in *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with*

High Intelligence and Emotional Quotients (Cat. No.99CH36289), vol. 2, Oct 1999, pp. 1087–1093 vol.2.

[35] L. Paletta and E. Rome, “Learning fusion strategies for visual object detection,” in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000) (Cat. No.00CH37113)*, vol. 2, Oct 2000, pp. 1446–1452 vol.2.

[36] M. Kolesnik and G. Baratoff, “3d interpretation of sewer circular structures,” in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, April 2000, pp. 1453–1458 vol.2.

[37] O. Moselhi and T. Shehab-Eldeen, “Automated detection of surface defects in water and sewer pipes,” *Automation in Construction*, vol. 8, no. 5, pp. 581 – 588, 1999.

[38] ——, “Classification of defects in sewer pipes using neural networks,” *Journal of Infrastructure Systems*, vol. 6, no. 3, pp. 97–104, 2000.

[39] T. Shehab-Eldeen and O. Moselhi, “Automated inspection of utility pipes: A solution strategy for data management,” in *19th International Symposium on Automation and Robotics in Construction (ISARC), Washington, U.S.A.*, 2002, pp. 531–536.

[40] T. Shehab and O. Moselhi, “Automated detection and classification of infiltration in sewer pipes,” *Journal of Infrastructure Systems*, vol. 11, no. 3, pp. 165–171, 2005.

[41] M. Browne, S. Shiry, M. Dorn, and R. Ouellette, “Visual feature extraction via pca-based parameterization of wavelet density functions,” in *International Symposium on Robots and Automation*, 2002, pp. 398–402. [Online]. Available: https://www.semanticscholar.org/paper/Visual-feature-extraction-via-PCA-based-of-wavelet-Browne-Shiry/642bb6b7a1f8460aa104faba6b47197ea2e02628

[42] M. Browne, M. Dorn, R. Ouellette, T. Christaller, and S. Shiry, “Wavelet entropy-based feature extraction for crack detection in sewer pipes,” in *6th International Conference on Mechatronics Technology, Kitakyushu, Japan*, 2002, pp. 202–206. [Online]. Available: https://www.semanticscholar.org/paper/Wavelet-Entropy-based-Feature-Extraction-for-Crack-Browne-Dorn/2c417cf5b746a39a9e7256257cdb4a8fb0e8a200

[43] M. Browne and S. S. Ghidary, “Convolutional neural networks for image processing: An application in robot vision,” in *AI 2003: Advances in Artificial Intelligence*, T. T. D. Gedeon and L. C. C. Fung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 641–652.

[44] R. Oullette, M. Browne, and K. Hirasawa, "Genetic algorithm optimization of a convolutional neural network for autonomous crack detection," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 1, June 2004, pp. 516–521 Vol.1.

[45] M. Browne, S. S. Ghidary, and N. M. Mayer, "Convolutional neural networks for image processing with applications in mobile robotics," in *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks*, B. Prasad and S. R. M. Prasanna, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 327–349.

[46] M.-D. Yang, T.-C. Su, N.-F. Pan, and P. Liu, "Feature extraction of sewer pipe failures by wavelet transform and co-occurrence matrix," in *2008 International Conference on Wavelet Analysis and Pattern Recognition*, vol. 2, Aug 2008, pp. 579–584.

[47] M.-D. Yang and T.-C. Su, "Automated diagnosis of sewer pipe defects based on machine learning approaches," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1327 – 1337, 2008.

[48] ——, "Segmenting ideal morphologies of sewer pipe defects on cctv images for automated diagnosis," *Expert Systems with Applications*, vol. 36, no. 2, Part 2, pp. 3562 – 3573, 2009.

[49] M. D. Yang, T. C. Su, N. F. Pan, and P. Liu, "Sewerage rehabilitation planning," in *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, Dec 2010, pp. 621–625.

[50] M.-D. Yang, T.-C. Su, N.-F. Pan, and P. Liu, "Feature extraction of sewer pipe defects using wavelet transform and co-occurrence matrix," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 09, no. 02, pp. 211–225, 2011.

[51] T.-C. Su, M.-D. Yang, T.-C. Wu, and J.-Y. Lin, "Morphological segmentation based on edge detection for sewer pipe defects on cctv images," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 094 – 13 114, 2011.

[52] T.-C. Su and M.-D. Yang, "Application of morphological segmentation to leaking defect detection in sewer pipelines," *Sensors*, vol. 14, no. 5, pp. 8686–8704, 2014.

[53] T.-C. Su, "Segmentation of crack and open joint in sewer pipelines based on cctv inspection images," in *2015 AASRI International Conference on Circuits and Systems (CAS 2015)*. Atlantis Press, 2015, pp. 263–266.

[54] D. Bairaktaris, V. Delis, C. Emmanouilidis, S. Frondistou-Yannas, K. Gratsias, V. Kallidromitis, and N. Rerras, "Decision-support system for the rehabilitation

of deteriorating sewers," *Journal of Performance of Constructed Facilities*, vol. 21, no. 3, pp. 240–248, 2007.

[55] P. Swarnalatha, M. Kota, N. R. Resu, and G. Srivasanth, "Automated assessment tool for the depth of pipe deterioration," in *2009 IEEE International Advance Computing Conference*, March 2009, pp. 721–724.

[56] W. Xue-Fei and B. Hua, "Automated assessment of buried pipeline defects by image processing," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, Nov 2009, pp. 583–587.

[57] N. Sarshar, M. Halfawy, and J. Hengmeechai, "Video processing techniques for assisted cctv inspection and condition rating of sewers," *Journal of Water Management Modeling*, vol. 17, pp. 129–147, 2009.

[58] M. R. Halfawy and J. Hengmeechai, "Efficient algorithm for crack detection in sewer images from closed-circuit television inspections," *Journal of Infrastructure Systems*, vol. 20, no. 2, p. 04013014, 2014.

[59] ——, "Automated defect detection in sewer closed circuit television images using histograms of oriented gradients and support vector machine," *Automation in Construction*, vol. 38, pp. 1 – 13, 2014.

[60] ——, "Optical flow techniques for estimation of camera motion parameters in sewer closed circuit television inspection videos," *Automation in Construction*, vol. 38, pp. 39 – 45, 2014.

[61] ——, "Integrated vision-based system for automated defect detection in sewer closed circuit television inspection videos," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, p. 04014024, 2015.

[62] A. Chaki and T. Chattopadhyay, "An intelligent fuzzy multifactor based decision support system for crack detection of underground sewer pipelines," in *2010 10th International Conference on Intelligent Systems Design and Applications*, Nov 2010, pp. 1471–1475.

[63] I. Khalifa, A. E. Aboutabl, and G. S. A. Barakat, "A new image-based model for predicting cracks in sewer pipes," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 12, pp. 65–71, 2013.

[64] I. Khalifa, A. E. Aboutabl, and G. S. A. A. Barakat, "A new image model for predicting cracks in sewer pipes based on time," *International Journal of Computer Applications*, vol. 87, no. 9, pp. 25–32, February 2014.

[65] P. Huynh, R. Ross, A. Martchenko, and J. Devlin, "Dou-edge evaluation algorithm for automatic thin crack detection in pipelines," in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Oct 2015, pp. 191–196.

[66] ——, “3d anomaly inspection system for sewer pipes using stereo vision and novel image processing,” in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, June 2016, pp. 988–993.

[67] J. Myrans, Z. Kapelan, R. Everson, and J. Britton, “Using support vector machines to identify faults in sewer pipes from cctv surveys,” in *Computing & Control for the Water Industry, Electronic Proceedings*, 2016.

[68] J. Myrans, Z. Kapelan, and R. Everson, “Automated detection of faults in wastewater pipes from cctv footage by using random forests,” *Procedia Engineering*, vol. 154, pp. 36 – 41, 2016, 12th International Conference on Hydroinformatics (HIC 2016) - Smart Water for the Future.

[69] ——, “Automatic detection of sewer faults using continuous cctv footage,” in *Computing & Control for the Water Industry*, 9 2017.

[70] ——, “Automatic identification of sewer fault types using cctv footage,” in *HIC 2018. 13th International Conference on Hydroinformatics*, ser. EPiC Series in Engineering, G. L. Loggia, G. Freni, V. Puleo, and M. D. Marchis, Eds., vol. 3. EasyChair, 2018, pp. 1478–1485.

[71] ——, “Using automatic anomaly detection to identify faults in sewers,” in *Water Distribution Systems Analysis / Computing & Control for the Water Industry Joint Conference 2018*, ser. WDSA / CCWI Joint Conference, vol. 1, 2018. [Online]. Available: https://ojs.library.queensu.ca/index.php/wdsa-ccw/article/view/12030

[72] J. Myrans, R. Everson, and Z. Kapelan, “Automated detection of faults in sewers using cctv image sequences,” *Automation in Construction*, vol. 95, pp. 64 – 71, 2018.

[73] J. Myrans, Z. Kapelan, and R. Everson, “Combining classifiers to detect faults in wastewater networks,” *Water Science and Technology*, vol. 77, no. 9, pp. 2184–2189, 03 2018.

[74] J. Myrans, R. Everson, and Z. Kapelan, “Automated detection of fault types in cctv sewer surveys,” *Journal of Hydroinformatics*, vol. 21, no. 1, pp. 153–163, 10 2018.

[75] S. Moradi and T. Zayed, “Real-time defect detection in sewer closed circuit television inspection videos,” in *Pipelines 2017*, 2017, pp. 295–307.

[76] S. Moradi, T. Zayed, and F. Golkhoo, “Automated sewer pipeline inspection using computer vision techniques,” in *Pipelines 2018*, 2018, pp. 582–587.

[77] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, “Automated defect classification in sewer closed circuit television inspections using

deep convolutional neural networks," *Automation in Construction*, vol. 91, pp. 273 – 283, 2018.

[78] S. S. Kumar and D. M. Abraham, "A deep learning based automated structural defect detection system for sewer pipelines," in *Computing in Civil Engineering 2019*, 2019, pp. 226–233.

[79] M. Wang and J. C. P. Cheng, "Development and improvement of deep learning based automated defect detection for sewer pipe inspection using faster r-cnn," in *Advanced Computing Strategies for Engineering*, I. F. C. Smith and B. Domer, Eds. Cham: Springer International Publishing, 2018, pp. 171–192.

[80] J. C. Cheng and M. Wang, "Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques," *Automation in Construction*, vol. 95, pp. 155 – 171, 2018.

[81] M. Wang and J. C. Cheng, "Semantic segmentation of sewer pipe defects using deep dilated convolutional neural network," in *26th International Symposium on Automation and Robotics in Construction (ISARC), Banff, Alberta, Canada*, 2019, pp. 586–594.

[82] M. Wang and J. C. P. Cheng, "A unified convolutional neural network integrated with conditional random field for pipe defect segmentation," *Computer-Aided Civil and Infrastructure Engineering*, pp. 1–15, 2019.

[83] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, "Deep learning-based automated detection of sewer defects in cctv videos," *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[84] L. M. Dang, S. I. Hassan, S. Im, I. Mehmood, and H. Moon, "Utilizing text recognition for the defects extraction in sewers cctv inspection videos," *Computers in Industry*, vol. 99, pp. 96 – 109, 2018.

[85] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, "Underground sewer pipe condition assessment based on convolutional neural networks," *Automation in Construction*, vol. 106, p. 102849, 2019.

[86] K. Chen, H. Hu, C. Chen, L. Chen, and C. He, "An intelligent sewer defect detection method based on convolutional neural network," in *2018 IEEE International Conference on Information and Automation (ICIA)*, Aug 2018, pp. 1301–1306.

[87] R. Tennakoon., R. Hoseinnezhad., H. Tran., and A. Bab-Hadiashar., "Visual inspection of storm-water pipe systems using deep convolutional neural networks," in *Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics - Volume 1: ICINCO,*, INSTICC. SciTePress, 2018, pp. 135–140.

[88] X. Ye, J. Zuo, R. Li, Y. Wang, L. Gan, Z. Yu, and X. Hu, "Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern chinese city," *Frontiers of Environmental Science & Engineering*, vol. 13, no. 2, p. 17, Jan 2019.

[89] D. Li, A. Cong, and S. Guo, "Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification," *Automation in Construction*, vol. 101, pp. 199 – 208, 2019.

[90] G. Heo, J. Jeon, and B. Son, "Crack automatic detection of cctv video of sewer inspection with low resolution," *KSCE Journal of Civil Engineering*, vol. 23, no. 3, pp. 1219–1227, Mar 2019.

[91] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Automation in Construction*, vol. 109, p. 102967, 2020.

[92] D. M. Abraham, M. J. Chae, and S. Gokhale, "Utilizing neural networks for condition assessment of sanitary sewer infrastructure," in *17th International Symposium on Automation and Robotics in Construction (ISARC), Taipei, Taiwan*, 2000, pp. 1–6.

[93] M. J. Chae and D. M. Abraham, "Automated condition assessment of sanitary sewer pipelines," in *Computing in Civil and Building Engineering (2000)*, 2000, pp. 1196–1203.

[94] ——, "Neuro-fuzzy approaches for sanitary sewer pipeline condition assessment," *Journal of Computing in Civil Engineering*, vol. 15, no. 1, pp. 4–14, 2001.

[95] S. K. Sinha, F. Karray, and P. W. Fieguth, "Underground pipe cracks classification using image analysis and neuro-fuzzy algorithm," in *Proceedings of the 1999 IEEE International Symposium on Intelligent Control Intelligent Systems and Semiotics (Cat. No.99CH37014)*, Sep. 1999, pp. 399–404.

[96] S. K. Sinha, P. W. Fieguth, M. A. Polak, and R. A. McKim, "Automated underground pipe condition assessment by image analysis of the state-of-the-art sewer scanner and evaluation technology surveys," in *No Dig '99 International Conference*, 1999. [Online]. Available: https://www.semanticscholar.org/paper/AUTOMATED-UNDERGROUND-PIPE-CONDITION-ASSESSMENT-BY-Sinha-Fieguth/e9b39e55ac1f969bf1c185212b51c72715afcb5d

[97] ——, "Novel system for automatic underground pipe distress detection and classification," in *Annual Conference of the Canadian Society for Civil Engineering*, 1999, pp. 279–288.

[98] R. A. McKim and S. K. Sinha, “Condition assessment of underground sewer pipes using a modified digital image processing paradigm,” *Tunnelling and Underground Space Technology*, vol. 14, pp. 29 – 37, 1999.

[99] P. W. Fieguth and S. K. Sinha, “Automated analysis and detection of cracks in underground scanned pipes,” in *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*, vol. 4, Oct 1999, pp. 395–399 vol.4.

[100] S. K. Sinha, P. W. Fieguth, and M. A. Polak, “Computer vision techniques for inspection of pipes,” in *Computing in Civil and Building Engineering (2000)*, 2000, pp. 418–425.

[101] S. K. Sinha, P. W. Fieguth, and F. Karray, “Automated segmentation of underground pipe scanned images,” in *World Automation Congress 2000*, Hawaii, 2000.

[102] S. K. Sinha, “Automated condition assessment of buried sewer pipeline using computer vision techniques,” in *Pipelines 2001*, 2001, pp. 1–12.

[103] S. K. Sinha and P. W. Fieguth, “Development of an automated pipeline inspection system,” in *Underground Infrastructure Research: Municipal, Industrial and Environmental Applications*, 2001, pp. 279–288.

[104] S. K. Sinha and F. Karray, “Classification of underground pipe scanned images using feature extraction and neuro-fuzzy algorithm,” *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 393–401, March 2002.

[105] S. K. Sinha, P. W. Fieguth, and M. A. Polak, “Computer vision techniques for automatic structural assessment of underground pipes,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 18, no. 2, pp. 95–112, 2003.

[106] S. K. Sinha and M. A. Knight, “Intelligent system for condition monitoring of underground pipelines,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, no. 1, pp. 42–53, 2004.

[107] S. K. Sinha and P. W. Fieguth, “Automated detection of cracks in buried concrete pipe images,” *Automation in Construction*, vol. 15, no. 1, pp. 58 – 72, 2006.

[108] ——, “Neuro-fuzzy network for the classification of buried pipe defects,” *Automation in Construction*, vol. 15, no. 1, pp. 73 – 83, 2006.

[109] ——, “Segmentation of buried concrete pipe images,” *Automation in Construction*, vol. 15, no. 1, pp. 47 – 57, 2006.

[110] ——, “Morphological segmentation and classification of underground pipe images,” *Machine Vision and Applications*, vol. 17, no. 1, p. 21, Jan 2006.

[111] S. Iyer and S. K. Sinha, “Automated condition assessment of buried sewer pipes based on digital imaging techniques,” *Journal of Indian Institute of Science*, vol. 85, pp. 235–252, 2005. [Online]. Available: https://www.semanticscholar.org/paper/Automated-condition-assessment-of-buried-sewer-on-Iyer-Sinha/34f86b6fdeab6b675c41345e115853b2590115f7

[112] ——, “A robust approach for automatic detection and segmentation of cracks in underground pipeline images,” *Image and Vision Computing*, vol. 23, no. 10, pp. 921 – 933, 2005.

[113] ——, “Segmentation of pipe images for crack detection in buried sewers,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 21, no. 6, pp. 395–410, 2006.

[114] W. Wu, Z. Liu, and Y. He, “Classification of defects with ensemble methods in the automated visual inspection of sewer pipes,” *Pattern Analysis and Applications*, vol. 18, no. 2, pp. 263–276, May 2015.

[115] K. Müller and B. Fischer, “Objective condition assessment of sewer systems,” in *2nd Leading Edge Conference on Strategic Asset Management*, 2007, pp. 641–652. [Online]. Available: https://www.semanticscholar.org/paper/Objective-Condition-Assessment-of-Sewer-Systems-M%C3%BCller-Fischer/69b3ab9ac9cba9462d201e6c456132165f3220e5

[116] S. Kirstein, K. Müller, M. Walecki-Mingers, and T. M. Deserno, “Robust adaptive flow line detection in sewer pipes,” *Automation in Construction*, vol. 21, pp. 24 – 31, 2012.

[117] J. Mashford, P. Davis, and M. Rahilly, “Pixel-based colour image segmentation using support vector machine for automatic pipe inspection,” in *AI 2007: Advances in Artificial Intelligence*, M. A. Orgun and J. Thornton, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 739–743.

[118] J. Mashford, M. Rahilly, P. Davis, and S. Burn, “A morphological approach to pipe image interpretation based on segmentation by support vector machine,” *Automation in Construction*, vol. 19, no. 7, pp. 875 – 883, 2010.

[119] J. Mashford, M. Rahilly, and D. Marney, “Processing by svm of haar wavelet transforms for discontinuity detection,” in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 2011. [Online]. Available: https://www.semanticscholar.org/paper/Processing-by-SVM-of-Haar-Wavelet-Transforms-for-Mashford-Rahilly/99eb912d56ac4f631c70cf462469b52220e6169d

[120] J. Mashford, M. Rahilly, B. Lane, D. Marney, and S. Burn, “Edge detection in pipe images using classification of haar wavelet transforms,” *Applied Artificial Intelligence*, vol. 28, no. 7, pp. 675–689, 2014.

[121] H. Ganegedara, D. Alahakoon, J. Mashford, A. Paplinski, K. Müller, and T. M. Deserno, “Self organising map based region of interest labelling for automated defect identification in large sewer pipe image collections,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, June 2012, pp. 1–8.

[122] J. Kunzel, T. Werner, P. Eisert, and J. Waschnewski, “Automatic analysis of sewer pipes based on unrolled monocular fisheye images,” in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2018, pp. 2019–2027.

[123] C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, “A vision-based system for internal pipeline inspection,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3289–3299, June 2019.

[124] S. Shiry and M. Browne, “Convolutional neural networks for robot vision: numerical studies and implementation on a sewer robot,” in *Proceedings of the 8th Australian and New Zealand Intelligent Information Systems Conference*, 2003, pp. 653–665. [Online]. Available: https://www.semanticscholar.org/paper/Convolutional-neural-networks-for-robot-vision-%3A-on-Shiry-Browne/15594b0e3061f5952aed8cfea8d0871ccdc0f210

[125] A. Ahrary and M. Ishikawa, “Detecting pipe feature points for sewer pipe system based on image information,” in *ICMIT 2005: Information Systems and Signal Processing*, vol. 6041, 2006.

[126] A. Ahrary, M. Ishikawa, and M. Okada, “Experimental evaluation of intelligent fault detection system for inspection of sewer pipes,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 1248–1253.

[127] A. Ahrary, Y. Kawamura, and M. Ishikawa, “An automated intelligent fault detection system for inspection of sewer pipes,” *IEEJ Transactions on Electronics, Information and Systems*, vol. 127, no. 6, pp. 943–950, 2007.

[128] W. Guo, L. Soibelman, and J. J. H. Garrett, “Automatic visual data interpretation for pipeline infrastructure assessment,” in *Responding to tomorrow’s challenges in structural engineering : IABSE symposium*, 2006.

[129] ——, “Automatic defect detection and recognition for asset condition assessment: A case study on sewer pipeline infrastructure system,” in *Computing in Civil Engineering (2007)*, 2007, pp. 419–426.

[130] W. Guo, L. Soibelman, and J. H. Garrett, “Imagery enhancement and interpretation for remote visual inspection of aging civil infrastructure,” *Tsinghua Science and Technology*, vol. 13, no. S1, pp. 375–380, Oct 2008.

[131] W. Guo, L. Soibelman, and J. Garrett, “Automated defect detection for sewer pipeline inspection and condition assessment,” *Automation in Construction*, vol. 18, no. 5, pp. 587 – 596, 2009.

[132] W. Guo, L. Soibelman, and J. H. Garrett, “Visual pattern recognition supporting defect reporting and condition assessment of wastewater collection systems,” *Journal of Computing in Civil Engineering*, vol. 23, no. 3, pp. 160–169, 2009.

[133] W. Guo, L. Soibelman, and J. J. H. Garrett, “Automated defect detection in urban wastewater pipes using invariant features found in video images,” in *Building a Sustainable Future*, 2009, pp. 1194–1203.

[134] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, “A defect classification methodology for sewer image sets with convolutional neural networks,” *Automation in Construction*, vol. 104, pp. 281 – 298, 2019.

[135] A. Hawari, M. Alamin, F. Alkadour, M. Elmasry, and T. Zayed, “Automated defect detection tool for closed circuit television (cctv) inspected sewer pipelines,” *Automation in Construction*, vol. 89, pp. 99 – 109, 2018.

[136] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, “Automatic detection and classification of sewer defects via hierarchical deep learning,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2019.

[137] M. Bertini, A. D. Bimbo, and L. Seidenari, “Multi-scale and real-time non-parametric approach for anomaly detection and localization,” *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 320 – 329, 2012, special issue on Semantic Understanding of Human Behaviors in Image Sequences.

[138] E. Epaillard and N. Bouguila, “Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas,” *Pattern Recognition*, vol. 55, pp. 125 – 136, 2016.

[139] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[140] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, “Machine learning interpretability: A survey on methods and metrics,” *Electronics*, vol. 8, no. 8, pp. 1–34, 2019.

[141] M. J. Chae, “Automated interpretation and assessment of sewer pipeline infrastructure,” Ph.D. dissertation, Purdue University, 2001, URL: https://search.proquest.com/docview/304724315.

[142] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[143] B. Frenay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, May 2014.

[144] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 233–240.

[145] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

Paper B

Sewer-ML: A Multi-Label Sewer Defect Classification Dataset and Benchmark

Joakim Bruslund Haurum and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 13451–13462, 2021.

The layout has been revised.

Abstract

Perhaps surprisingly sewerage infrastructure is one of the most costly infrastructures in modern society. Sewer pipes are manually inspected to determine whether the pipes are defective. However, this process is limited by the number of qualified inspectors and the time it takes to inspect a pipe. Automatization of this process is therefore of high interest. So far, the success of computer vision approaches for sewer defect classification has been limited when compared to the success in other fields mainly due to the lack of public datasets. To this end, in this work we present a large novel and publicly available multi-label classification dataset for image-based sewer defect classification called Sewer-ML.

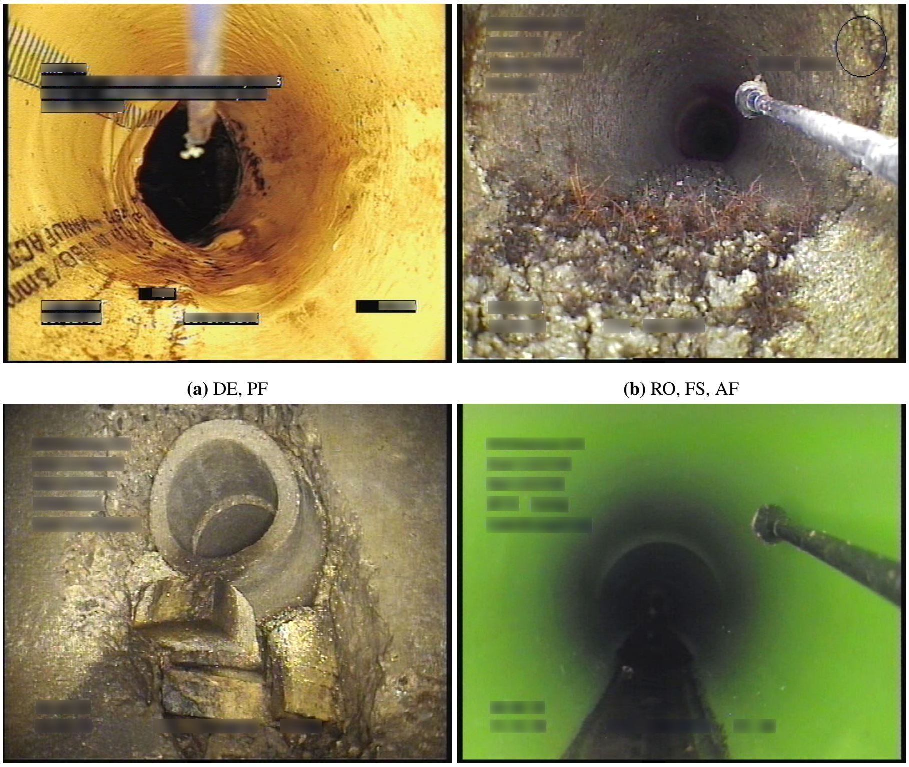
The Sewer-ML dataset consists of 1.3 million images annotated by professional sewer inspectors from three different utility companies across nine years. Together with the dataset, we also present a benchmark algorithm and a novel metric for assessing performance. The benchmark algorithm is a result of evaluating 12 state-of-the-art algorithms, six from the sewer defect classification domain and six from the multi-label classification domain, and combining the best performing algorithms. The novel metric is a class-importance weighted F2 score, $F2_{CIW}$, reflecting the economic impact of each class, used together with the normal pipe F1 score, $F1_{Normal}$. The benchmark algorithm achieves an $F2_{CIW}$ score of 55.11% and $F1_{Normal}$ score of 90.94%, leaving ample room for improvement on the Sewer-ML dataset. The code, models, and dataset are available at the project page `http://vap.aau.dk/sewer-ml`

1 Introduction

The sewerage infrastructure is an important but often unnoticed infrastructure. 240 million US citizens are serviced by 1.28 million kilometers of public sewer pipes and 800,000 kilometers of privately owned pipes [1]. In order to maintain public health and sanitation, and avoid *e.g.* unintentional sewer overflows, a 271 billion dollar investment is needed within the next 10 years in order to service an additional 56 million US citizens [1]. Additionally, all of these sewer pipes have to be regularly inspected to avoid sudden pipe collapse or reduced sewer capabilities.

Sewer inspections are currently performed on location by a professional inspector, who simultaneously maneuvers a remote controlled vehicle with a movable camera through the sewer pipe. This is hard and tiresome work, as the inspectors must look at a video feed for a prolonged amount of time. This can lead to flawed inspections, which in the worst case can result in damage to the sewerage infrastructure. Furthermore, the variance in visual appearance within sewer pipes further complicates the task, see Figure B.1.

Therefore, the field of automated sewer inspection has been researched by industry and academia for the last three decades, through the development of different robot platforms and specialized algorithms [2]. However, there are at the moment no means



(a) DE, PF

(b) RO, FS, AF

(c) PH, RB

(d) Normal pipe

Fig. B.1: Sewer-ML data examples. Images showcasing a subset of the classes and the visual variation in the dataset. The class codes below each image are described in Table B.1.

to determine which method is the best. Haurum and Moeslund [2] found that there are no open-source benchmark datasets, little to no open-source code, and no agreed upon metrics or evaluation protocol. Instead, many researchers utilize their own datasets from different countries and follow different inspection guides. This leads to stagnation in the field when compared to other computer vision fields and a lack of reproducibility in the automated sewer inspection field.

For these reasons, we present the open-source *Sewer-ML* multi-label defect dataset, containing 1.3 million images annotated by professional sewer inspectors. The dataset is collected from three different Danish water utility companies over a period of nine years. Our contributions are fourfold:

- A publicly available multi-label sewer inspection dataset with 1.3 million annotated images.
- An open-source comparison of state-of-the-art methods using the new dataset.
- A novel, class-importance weighted F2 metric, F2_{CIW}.

- A benchmark algorithm combining knowledge from sewer defect and multi-label classification domains.

The paper is structured as follows. In Section 2, we review the related works within the multi-label image classification and automated sewer inspection fields. In Section 3, the proposed dataset is introduced and described in detail. In Section 4, we introduce our novel metric, test several state-of-the-art methods on the new dataset, and conduct an ablation study on the obtained results leading to our benchmark algorithm. Finally, in Section 5, we summarize our findings and conclude the paper.

2 Related Works

Multi-label Image Classification. Through the years, the field of multi-label classification has experienced several different trends. Classically, the naive way to approach the problem has been to use an ensemble of binary classifiers and ignore label correlations [3]. This approach has been replaced by methods consisting of a single model incorporating the label correlations into the method itself. These trends have included ranking the label predictions [4–6], utilizing object localization techniques and attention mechanisms [7–16], or incorporating a recurrent sub-network to encode label dependencies [10, 12, 17–20].

Current state-of-the-art networks focus on utilizing the inherent graph nature of the multi-label problem [21–27], by using the co-occurrence matrix between labels in combination with graph convolutional networks (GCNs) [28]. Chen *et al.* [23] proposed the ML-GCN method, which combines the output of a two-layer GCN with the last feature map of a ResNet-101 [29] network to achieve a well performing multi-label classifier. Wang *et al.* [26] built upon this idea in their KSSNet model. KSSNet improves the performance over ML-GCN by fusing features from a GCN into the final feature map of each residual block in a ResNet-101 model, using a novel lateral connection module. Furthermore, the GCN adjacency matrix is created by combining the label correlation matrix with a label knowledge graph. Lastly, it is also possible to simply take a network which has been proven to work well on a multi-class classification task and instead train it with a relevant loss objective, often the binary cross-entropy loss. This is the case with the recent work of Wu *et al.* [30] who utilized the ResNet-101 architecture and Ridnik *et al.* [31] who proposed a modified variation of the ResNet architecture, called TResNet. The TResNet network has outperformed several models designed for the multi-label task.

The multi-label image classification field has classically worked on smaller datasets such as PASCAL VOC [32], NUS-WIDE [33], and COCO [34], each containing between 5 to 80 thousand training images and 20-80 classes. Therefore, the applied methods have often relied on pre-training the backbone network on ImageNet [35]. However, recently the Tencent-ML [30] and Open Images datasets [36], each containing between approximately 6 and 12 million training images and 11 to 20 thousand classes, have been proposed. These datasets allow for training methods directly on the multi-label task and not pretraining on ImageNet. All of these datasets focus on natural scene images with "common" objects. Furthermore, these datasets are often severely imbalanced, as *e.g.* the class "person" occurs more frequently than the class "sheep". Therefore, there have been attempts to counteract the data imbalance through weighting the loss objective. This has classically been achieved by utilizing some variant of the inverse class frequency, though custom loss objectives have been proposed specifically for the imbalanced data problem [37–40].

Automated Sewer Inspection. For several decades there has been an increasing industrial and academic interest in automating the sewer inspection process. This line of research builds heavily upon the general computer vision field, though the current state-of-the-art does not fully utilize the recent advances within computer vision.

Several different types of sensors [41, 42], such as acoustic sensors [43–45], laser scanners [46, 47], and depth sensors [48–51], have all been utilized for sewer pipe reconstruction and detecting specific defects but have not seen widespread usage in more generalized tasks. Conversely, image and video based approaches have been utilized to detect, segment, and classify a wide variety of sewer defects. Traditionally, hand-crafted features and small, model based classifiers or heuristic decision rules have been utilized [52–54]. However, in recent years deep learning based methods have gained traction within the field. This has led to advances within video processing [55–58], water level estimation [59, 60], defect detection [61–63], segmentation [64–66], and classification in multi-class and multi-label settings [52, 67–72]. For a full review of the field we refer to Haurum and Moeslund [2].

Within sewer defect classification there has been a recent increase in interest, focused on three different system settings: a single end-to-end classifier, a two-stage approach consisting of a binary classifier and a multi-class/label classifier, and an ensemble of binary classifiers. Kumar *et al.* [69] utilized an ensemble of binary classifiers to categorize four types of defects using a small, two-layer CNN trained in a one vs. all manner. Hassan *et al.* [68] used AlexNet [73] and Li *et al.* [70] a modified ResNet-18 network [29], trained in an end-to-end manner. Similarly, Meijer *et al.* [71] built upon the work of Kumar *et al.* using a small, three-layer CNN for multi-label defect classification, trained end-to-end. Lastly, Xie *et al.* [72], Chen *et al.* [67], and Myrans *et al.* [52] all used two-stage approaches. Xie *et al.* trained two small, three-layer CNNs, where the first CNN determines whether a defect is present, while the second CNN, a fine-tuned version of the first, classifies the defects. Chen *et al.*, on the other hand, use the lightweight SqueezeNet [74] network for the initial binary defect classification and the deeper InceptionV3 [75] network for predicting the defect class. Myrans *et al.* differ from the other recent methods by using the GIST feature descriptor [76] and two Extra Trees [77] classifiers in sequence.

All prior methods utilize separate private dataset with different classes and class distributions, due to the inherent commercial interest involved in the field [2]. The datasets are typically either balanced such that the number of observations per class is balanced, the number of normal and defect observations are balanced, or the dataset is not balanced but inherently skewed. For example, Meijer *et al.* [71] utilized a dataset consisting of 2.2 million images, but only 17,663 of those images contain defects. In order to counteract this large imbalance, Meijer *et al*. increased the number of defective observations by a factor of five through oversampling. Additionally, there are no common metric nor evaluation protocol, making fair comparison between methods impossible [2]. All of these factors severely hinder the reproducibility and progress within the field.

3 The Sewer-ML Dataset

In this section, we present how the data was collected (Section 3.1), how the multi-label ground truth annotations are obtained (Section 3.2), how the dataset is constructed (Section 3.3), and how we redact information which is present in the images (Section 3.4). Further dataset insights are presented in the supplementary materials.

3.1 Data Collection

A total of 75,618 annotated sewer inspection videos were obtained from three different Danish water utility companies from the period 2011–2019. All videos were annotated by licensed sewer inspectors following a common Danish standard [78] containing 18 specific classes listed in Table B.1. According to the inspection standard, each class is given a point score representing the economic consequence of the class, which is determined by professionals involved in the sewer inspection field [79]. We normalize the point scores to the interval $[0,1]$ by dividing all point scores by the largest one, denoting the new values as the *class-importance weight* (CIW). The collected data span a large variety of materials, shapes, and dimensions from both main and lateral pipes. This leads to a large variety in the available data, reflecting the natural variance observed during actual sewer inspections.

3.2 Multi-Label Ground Truth

The dataset is constructed by extracting a single frame at each class annotation in a sewer inspection video. Each annotation corresponds to a ground truth annotation of a single class at a specific second in the video, with an associated location within the pipe. We obtain the multi-label representation by combining annotations close to each other in the pipe. This is a noisy approach as the camera can rotate in a hemisphere and does not guarantee that all annotations will be visible. For each annotation in an inspection video, we aggregate the annotated class with all other annotated classes which are up to 0.3 meters earlier in the pipe or 1.0 meters ahead in the pipe. These values have been decided through manual inspection as the position measurement can be noisy. This is necessary in order to include nearby and upcoming, visible classes. Lastly, some entries are noted as *continuous*, which means the class occurs frequently within a specified stretch of the pipe, but are not explicitly annotated at each occurrence.

Table B.1: Sewer inspection classes. Overview and short description of each annotation class [78] and the class-importance weights (CIW) [79].

Code	Description	CIW
VA	Water Level (in percentages)	0.0310
RB	Cracks, breaks, and collapses	1.0000
OB	Surface damage	0.5518
PF	Production error	0.2896
DE	Deformation	0.1622
FS	Displaced joint	0.6419
IS	Intruding sealing material	0.1847
RO	Roots	0.3559
IN	Infiltration	0.3131
AF	Settled deposits	0.0811
BE	Attached deposits	0.2275
FO	Obstacle	0.2477
GR	Branch pipe	0.0901
PH	Chiseled connection	0.4167
PB	Drilled connection	0.4167
OS	Lateral reinstatement cuts	0.9009
OP	Connection with transition profile	0.3829
OK	Connection with construction changes	0.4396

Table B.2: Split between defective and normal observations. Number of images containing normal and defective observations in the three dataset splits.

Type	Training	Validation	Test	Total
Normal	552,820	68,681	69,221	690,722
Defective	487,309	61,365	60,805	609,479
Total	1,040,129	130,046	130,026	1,300,201

We handle this edge case by adding the continuous class to all other annotated class occurrences within the defined pipe stretch.

The 18 classes are not all instances of pipe defects but can also indicate important information such as a change in pipe shape or material, occurrence of a branch pipe or pipe connections. The VA class is a special class, as it is annotated at the start and end of an inspection video, as well as when the water level changes within a 10% step interval. This means all annotations have an associated water level.

Additionally, we obtain observations of cases with no annotated classes, denoted non-defective (ND), using a set of heuristic rules. First, we apply a one meter buffer zone around each annotated class, such that there is at least two meters between annotated classes before ND images can be extracted. If there are any active continuous class between the annotated classes, no ND images are extracted. Furthermore, we

Table B.3: Sewer dataset comparison. A comparison of datasets used for sewer defect classification and the proposed Sewer-ML dataset. We report whether the dataset is publicly available (P), the annotations are multi-label (ML), the number of images with defects (DI), images with normal pipes (NI), annotated classes (C), and the Class Imbalance (CI) for each dataset rounded to the nearest integer.

Dataset	P	ML	DI	NI	C	CI
Ye *et al.* [54]			1,045	0	7	13
Myrans *et al.* [52]			2,260	0	13	102
Chen *et al.* [67]			8,000	10,000	5	5
Li *et al.* [70]			8,455	9,878	7	19
Kumar *et al.* [69]			11,000	1,000	3	4
Meijer *et al.* [71]		✓	17,663	2,184,919	12	12,732
Xie *et al.* [72]			22,800	20,000	7	8
Hassan *et al.* [68]			24,137	0	6	3
Sewer-ML	✓	✓	609,479	690,722	17	123

enforce that the inspection vehicle may at maximum move 0.25 m/s, calculated based on the time and distance difference between the two classes. This restriction is based on the maximum speed the inspectors are allowed to move the inspection vehicle during an inspection. Lastly, ND images are only extracted when the inspection vehicle is moving forward through the pipe. This condition is checked using the distance information associated with each annotation. If these conditions are met we can extract ND images. In order to avoid duplicate images of the same pipe area, we extract one ND image per meter uniformly sampled between the two annotated classes. The video timestamps of the ND images are calculated using a constant velocity assumption. Examples from the dataset are shown in Figure B.1 and the supplementary materials.

Moreover, the VA class is special, as it is a continuous entity throughout the video. The VA annotations are grouped together with the ND class if there are no other co-occurring labels. This leads to a total of 690,722 images of “normal” pipes with no annotated classes and 609,479 images with one or more annotated classes which we call “defective”, resulting in a total of 1,300,201 images. Lastly, we pose the multi-label classification problem as predicting the class labels in Table B.1, except for the VA class. This means a normal pipe with no class annotations is the absence of any classes. Therefore, it is an **implicit** class.

3.3 Dataset Construction

We construct the dataset by first splitting the data into three splits: training, validation and test. We randomly select videos until 80% of all annotations are in the training split and the remaining 20% equally split between the validation and test splits. This leads to 60,356 videos for training, 7,692 videos for validation, and 7,570 videos for testing. This way it is ensured that no images from the same pipe are present between splits. These splits lead to a near even split of normal and defective observations, see

Table B.2.

Looking at the distribution of the class occurrences, as shown in Figure B.2, the occurrences are evenly represented in each split, suggesting a similar class distribution in each of the splits. Moreover, it is evident that the constructed dataset is skewed towards a few major classes, such as the “Normal” and “FS” classes. This visually shows the large imbalance in the dataset, representative of the real life distribution of the classes. Unlike prior sewer inspection datasets, we do not manually balance the classes. We quantify the class imbalance (CI) in the dataset by calculating the ratio between the largest and smallest class and compare with the previously used sewer datasets, see Table B.3. Meijer *et al.* have a large CI due to sampling every five centimeters, resulting in a large number of normal images. Uniquely, Sewer-ML contains a large number of defect images, which are needed to train discriminative classifiers.

Similarly, it is interesting to see how often several classes are present at the same time. For each split, we plot the distribution of the number of labels in the observations in Figure B.2. In this plot we count the normal observations as having zero labels as it is an implicit label. We see that there is an equal number of observations with one or two classes and the number of observations reducing as more classes are present. We quantify this using the label cardinality (LC) using Equation B.1 [80] for each split. For these measures, we count the normal pipe observations as having one label.

$$\text{LC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C+1} y_c^{(i)} \tag{B.1}$$

where N is the number of observations in the split, C is the number of annotated classes, and $y_c^{(i)}$ is the ground truth value for class c in observation i.

We find that across splits, the LC is 1.49-1.50, indicating that on average there are 1.5 labels per observation. We cannot compare this with the LC of the datasets in Table B.3, as the datasets and ground truth data are not public.

3.4 Data Anonymization

The raw data provided by the water utility companies have all been post-processed by the inspection software to include metadata and annotation text information on the video itself. In order to avoid including ground truth information in the images and any potential privacy issues, the text has been redacted as shown in Figure B.3. Since the overlaid information is not static through the inspection video, due to *e.g.* class codes appearing on screen or pipe material changing, a single redacting mask cannot be used. This leads to a large annotation task, which would be long and tiresome to do manually. Instead, inspired by Borisyuk *et al.* [81], we train a Faster-RCNN [82] model on examples from the overlaid text data. 23,044 videos are used, with one frame extracted per video. The data is split into a training split of 20,739 images and a validation split of 2,305 images. All text information is manually annotated with

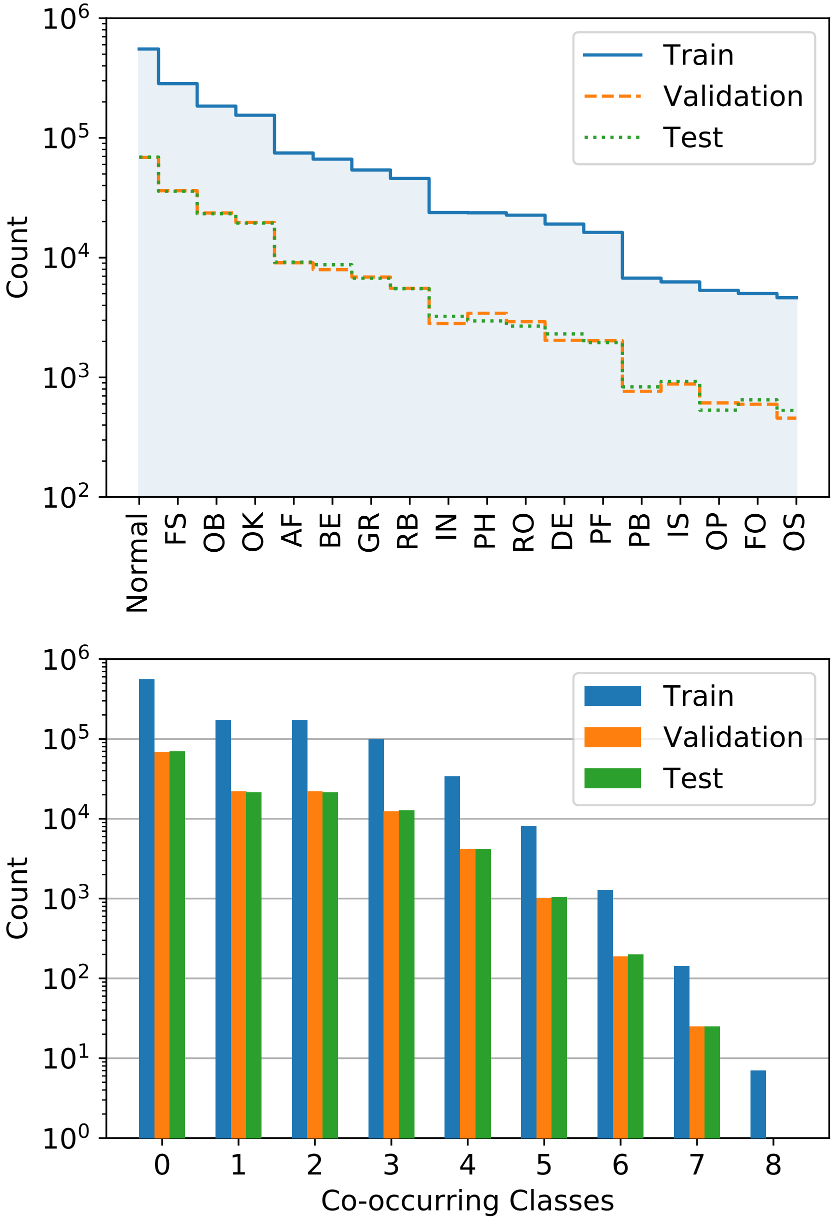
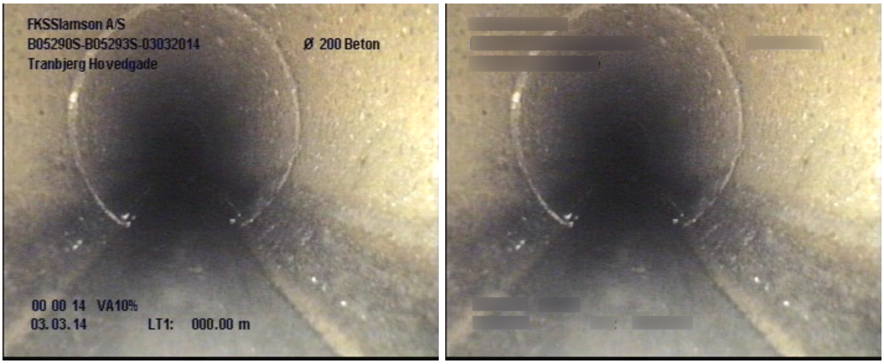


Fig. B.2: Dataset label statistics. The frequencies of the annotated classes and the normal class are shown in the top plot in descending order. The frequencies of the number of labeled classes per split are shown in the bottom plot, where “Normal” pipes have zero labeled classes. Note that the y-axes are log-scaled.



(a) Before text redaction.

(b) After text redaction.

Fig. B.3: Effect of the data anonymization process. By applying our text redaction pipeline the system is capable of detecting and blurring all text information on the images.

bounding boxes. The Faster-RCNN backbone is a ResNet-50 FPN [83] pre-trained on ImageNet [35]. We fine-tune the last three residual blocks. As the text data is distinctly different from the data present in the COCO dataset, we use custom anchor boxes. We choose to use three anchor box ratios and five scales, based on the bounding box ratio and area information from the training split. Full details are available in the supplementary materials.

Using the COCO metrics [34], we achieve an mAP@[0.75] of 96.39% and mAP@[0.5:0.95] of 89.10%. The Faster-RCNN model is applied on all 1.3 million images in the dataset, and the detected text is removed by applying a Gaussian blur kernel with a radius of 51 pixels. While this is not a perfect metric score, looking at the detections tells another story. We find that the model detects strings of text, annotated with several bounding boxes, as a single bounding box. An example of this is the “Ø 200” in Figure B.3. Similarly, text annotated with a single bounding box, are at times detected with several boxes. This leads to a lower metric score even though the redactions are correct. Therefore, we conclude that data leakage is not an issue in the dataset.

4 Benchmark

In this section, we present an approach that can be used as benchmarking for future work on the dataset. To this end, we first select (Section 4.1), train (Section 4.2), and test current state-of-the-art algorithms from the sewer defect classification and the general multi-label classification domains in order to see how they perform on the dataset (Section 4.4), using our novel class-importance weighted F2 metric (Section 4.3). Finally, we conduct an ablation study leading to our benchmark algorithm (Section 4.5), and discuss the per-class performance (Section 4.6).

4.1 Methods

From the sewer inspection domain we compare the methods proposed by Kumar *et al.* [69], Meijer *et al.* [71], Xie *et al.* [72], Chen *et al.* [67], Hassan *et al.* [68], and Myrans *et al.* [52]. These six methods are chosen to represent the recent advances within sewer defect classification [2]. For the ensemble of binary classifiers and the end-to-end methods, we train using the full dataset. However, for the two-stage classification approach, the first stage is trained on the full dataset to predict the presence of *any* annotated class, while the second stage is trained to predict the classes from a subset of the data containing annotated classes.

From the general multi-label classification domain, we choose four of the current best performing methods on the COCO and VOC datasets [37]. We choose two state-of-the-art graph-based methods, ML-GCN by Chen *et al.* [23] and KSSNet by Wang *et al.* [26], each utilizing a ResNet-101 [29] backbone. Furthermore, we also test the vanilla ResNet-101 model, as used by Wu *et al.* [30], and the TResNet architectures from Ridnik *et al.* [31], where we compare the medium, large, and extra-large versions of the model. All models are trained using an end-to-end approach.

While the normal/defect classification task is intrinsically related to the anomaly detection task, we do not compare with the state-of-the-art anomaly detection methods [84]. This is due to the sewer pipe owners requiring the defect classes in order to correctly manage their assets.

4.2 Training Procedure

Hyperparameters. In order to ensure comparability, we train all networks from scratch using the exact same training procedures. We base our training procedure on the methodology proposed by Goyal *et al.* [85] for efficiently training models on ImageNet. We train each network for 90 epochs, with a batch size of 256 using SGD with momentum. We utilize a learning rate of 0.1, momentum of 0.9, weight decay of 0.0001, and multiply the learning rate by 0.1 at epochs 30, 60, and 80. The ML-GCN and KSSNet networks utilize re-weighted correlation matrices in the GCN subnet, where the hyperparameters stated by the original authors are used. We do not construct a knowledge graph for KSSNet, as the class labels are abbreviations containing little semantic information. We use a one-hot encoding for the initial input to the GCN. For the Myrans *et al.* [52] system, the standard GIST hyperparameters are used and the first and second stage classifiers use 100 and 250 trees, respectively, a maximum depth of 10, and $\log_2(d)$ features when splitting nodes, where d is the dimensionality of the GIST feature vector. We find the hyperparameters through a small grid search, described in the supplementary materials.

Data augmentation. The training data are pre-processed by resizing the images to 224×224, horizontally flipping with a 50% chance, jittering the brightness, contrast, saturation, and hue by $\pm 10\%$ of the original values, and normalizing the data using the training split channel mean and standard deviation. During inference the images are simply resized to 224×224 and normalized. For the InceptionV3 network used by Chen *et al.*, the images are resized to 299×299 [75]. For the GIST features the images are converted to grayscale and resized to 128×128 [52].

Loss objective. We train using the standard binary cross-entropy loss, see Equation B.2, which is commonly used in the multi-label image classification domain.

$$L(\mathbf{x},\mathbf{y}) = \frac{1}{C}\sum_{c}^{C} -[w_c y_c \log(\sigma(x_c)) + (1-y_c)\log(1-\sigma(x_c))] \tag{B.2}$$

where C is the number of annotated classes in the dataset, y_c denotes whether class c is present in the current image, x_c is the raw output of the model for class c, σ is the sigmoid function, and w_c is the weight for class c if it is present in the current image.

As the dataset is imbalanced, we weight each positive class observation by the negative-to-positive class observation ratio, w_c, calculated using Equation B.3. This way the loss of minority classes are weighted higher when present in the images, while the loss of majority classes are weighted lower when present. For the InceptionV3 network, a lower weighted loss from the auxiliary classifier is added.

$$w_c = \frac{N - N_c}{N_c} \tag{B.3}$$

where N is the number of images in the training split, and N_c is the number of images in the split containing class c.

4.3 Metrics

Currently, there is no consensus on how sewer defect classification methods should be evaluated [2]. Commonly, the accuracy is used, but this is a poor metric when working with skewed datasets. Moreover, the metrics do not include domain knowledge. Therefore, we evaluate the model performance using two metrics incorporating domain knowledge, based on the Fβ metric [86],

$$\text{F}\beta = (1+\beta^2)\frac{\text{Prc}\cdot\text{Rcll}}{\beta^2\text{Prc}+\text{Rcll}} \tag{B.4}$$

where Prc and Rcll are the precision and recall of the classifier, respectively, and β is a weighting of recall, such that the recall β times more important than precision.

When performing sewer inspections, false negatives have a larger economic impact than false positives. This is due to false negatives possibly leading to faulty pipes going unnoticed, whereas a human will verify the predicted classes before a renovation decision is made. Therefore, it is more important to have a high recall than high precision, if both cannot be achieved. To incorporate this domain knowledge into the evaluation, we set $\beta = 2$ when evaluating the annotated classes. This is similar to previous tasks where recall is weighted higher than precision [87–89]. The per-class F2-scores are averaged using a novel, class-importance weighted F2-score, F2_{CIW}. The classes are weighted by the associated CIW, see Table B.1, as classes with a high CIW will be of larger importance for the pipe owners. F2_{CIW} is calculated as shown in Equation B.5.

$$\text{F2}_{\text{CIW}} = \frac{\sum_{c=1}^{C} \text{F2}_c \cdot \text{CIW}_c}{\sum_{c=1}^{C} \text{CIW}_c} \tag{B.5}$$

where CIW_c and F2_c are the CIW and F2-score for class c, respectively, and C is the number of annotated classes.

However, the normal pipes are not included in the F2_{CIW} computation, as normal pipes do not have a CIW. In order to quantify whether the tested methods can handle the absence of classes, and not simply maximize the F2_{CIW} score by predicting one or more classes at all times, we use the F1-score for the normal pipes, $\text{F1}_{\text{Normal}}$.

4.4 Model Performances

We report the validation and test split results of each model in Table B.4. Unless otherwise noted, a threshold of 0.5 is used to binarize the predictions. For the two-stage approaches, the prediction score from the first stage is used for all classes if the binary classifier detects no classes, and otherwise, the score of the second stage network is used. The results are obtained using the model weights from the epoch with the lowest validation loss. In most cases, the lowest validation loss is obtained after 30-40 epochs, whereafter the networks start overfitting. This indicates that while we utilize a dataset nearly the size of ImageNet, it might not be necessary to train for as long, due to all images being from the same visual domain. We also see that the small CNNs from Kumar *et al.* and Meijer *et al.* immediately diverge during training. This is possibly due to only applying two or three pooling layers before connecting to dense layers, leading to a parameter count of 269 and 135 million, respectively. Comparatively, the small CNN used by Xie *et al.*, uses three pooling layers as well as a pixel stride of two in the last two convolutional layers, leading to a parameter count of nine million parameters. Similarly, we observe that the ML-GCN method also diverges immediately, whereas the KSSNet method manages to train. We hypothesize that this is due to the lateral connections in the KSSNet adding stability during training. The loss curves are reported in the supplementary materials. We observe that the methods from the multi-label classification domain are better at classifying the specific classes, with TResNet-L achieving a F2_{CIW} test score of 54.75%. However, the simple two-stage approach by Xie *et al.* achieves the highest $\text{F1}_{\text{Normal}}$ score of 90.62%. This indicates that the approach by Xie *et al.* excels at distinguishing whether there are *any* classes, but not which one. The results are not solely due to the two-stage approach. Chen *et al.* also utilize a two-stage approach, but this produces significantly worse results. It is observed that the first stage simply predicts a “defect” in all images, which the later stage cannot properly handle. This is reflected by a low $\text{F1}_{\text{Normal}}$ score. Therefore, it appears there is value in using a small CNN for the first stage.

Table B.4: Performance metrics for each method. We present the different metrics for each method. The metrics are presented as percentages, and the highest score in each column is denoted in bold. The Kumar [69], Meijer [71] and ML-GCN [23] methods are not shown as they diverged during training. The "Sewer" and "General" identifiers indicate whether the method is from the sewer defect or multi-label classification domains, respectively. The classic multi-label metrics [24] are reported in the supplementary materials.

	Model	**Validation**		**Test**	
		F2$_{\text{CIW}}$ ↑	**F1$_{\text{Normal}}$ ↑**	**F2$_{\text{CIW}}$ ↑**	**F1$_{\text{Normal}}$ ↑**
Sewer	Xie [72]	48.57	**91.08**	48.34	**90.62**
	Chen [67]	42.03	3.96	41.74	3.59
	Hassan [68]	13.14	0.00	12.94	0.00
	Myrans [52]	4.01	26.03	4.11	27.48
General	ResNet-101 [29]	53.26	79.55	53.21	78.57
	KSSNet [26]	54.42	80.60	54.55	79.29
	TResNet-M [31]	53.83	81.23	53.79	79.91
	TResNet-L [31]	**54.63**	81.22	**54.75**	79.88
	TResNet-XL [31]	54.42	81.81	54.24	80.42

4.5 Ablation Studies and Benchmark Algorithm

Looking at the results in Table B.4, there is merit to both the end-to-end and two-stage approaches. We investigate whether the results can be improved further by combining end-to-end and two-stage methods. In the supplementary materials we report two additional ablation studies focused on getting a better understanding of the two-stage results.

Effect of different second stage classifiers. Based on our results in Table B.4, we look into whether combining the general multi-label methods with two-stage approaches would lead to state-of-the-art performance. Specifically, we combine the first stage of Xie *et al.* with each of the multi-label classifiers in Table B.4. The results are shown in Table B.5. We observe that by utilizing the first stage of Xie *et al.* both the F2$_{\text{CIW}}$ and F1$_{\text{Normal}}$ scores are improved when compared to the best results in Table B.4. Moreover, the performance is improved for all tested methods. Specifically, by using the first stage to filter out normal pipes, all general multi-label methods increase their F2$_{\text{CIW}}$ scores by approximately 0.5-1 percentage points, and the F1$_{\text{Normal}}$ by up to 10-12 percentage points. For the sewer domain methods their F2$_{\text{CIW}}$ scores are increased by 5-13 percentage points, and the F1$_{\text{Normal}}$ by 65-90 percentage points. From these results we can conclude that using a two-stage approach with the binary classifier from Xie *et al.* [72] and the TResNet-L model [31] is the *Benchmark* algorithm on Sewer-ML, with an F2$_{\text{CIW}}$ score of 55.11% and F1$_{\text{Normal}}$ score of 90.94%.

Table B.5: Two-stage classifier permutations. We evaluate each of the tested multi-label classifiers in a two-stage setup together with the first stage used by Xie *et al*. [72].

	Second Stage	**Validation**		**Test**	
		F2$_{\textbf{CIW}}$ ↑	**F1**$_{\textbf{Normal}}$ ↑	**F2**$_{\textbf{CIW}}$ ↑	**F1**$_{\textbf{Normal}}$ ↑
Sewer	Chen [67]	48.67	91.06	48.19	90.60
	Hassan [68]	18.08	91.08	17.89	90.62
	Myrans [52]	27.87	91.08	27.83	90.62
General	ResNet-101 [29]	54.45	91.28	54.01	90.88
	KSSNet [26]	**55.37**	91.30	55.09	**90.95**
	TResNet-M [31]	54.58	91.33	54.32	90.93
	TResNet-L [31]	55.36	91.32	**55.11**	90.94
	TResNet-XL [31]	54.97	**91.37**	54.51	**90.95**

4.6 Per-Class Performance

To gain a better understanding of the difficulty of detecting the different defects compared to their economical impact, we compare the F2 score for each defect with the corresponding CIW scores, see Figure B.4. We find that each of the classes with a high F2 score exhibit low intra-class and high inter-class variance, as well as more frequently occurring in the dataset. The displaced joint class FS exhibits limited intra-class variance due to limitations in where the defect can occur within the pipe, while being distinct from the other classes. Similarly, the surface damage class OB occurs frequently in the dataset and exhibits high inter-class variance due to the distinct visual appearance of the class.

Contrarily, the lower scoring defect classes exhibit a larger intra-class variance, lower inter-class variance, and are less frequently occurring. The obstacle class FO consists of a wide span of objects, *e.g*. a soda can, a leftover hammer, or another pipe which goes through the main pipe. The RB class exhibits large intra-class variance, due to the class encompassing cracks, breaks, and collapses, and low inter-class variance, due to the similarity in appearance between *e.g*. cracks and the fine roots in the RO class.

We observe that most of the lower scoring defects do not have a large economic impact. However, the two defects with the highest economic impact, OS and RB, are among the lowest scoring classes. Therefore, in order to improve the performance of the classification system, the detection rate on these two classes should be the main priority.

5 Conclusion

Sewerage infrastructure is a fundamental part of modern society and is continuously expanded. However, current manual inspections are tedious and slow when compared to the immense number of pipes that have to be inspected. Therefore, automated

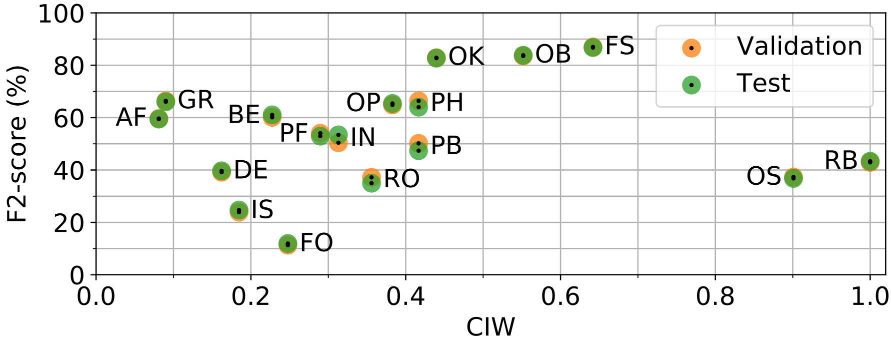


Fig. B.4: Per-class performance. Per-class F2 scores of the *Benchmark* algorithm (TResNet-L + Xie *et al.*), plotted against the corresponding CIW values.

sewer inspection technologies are crucial to ensuring the quality of our sewerage infrastructure. However, current state-of-the-art sewer defect classification methods have not yet adopted recent advances within computer vision. In order to facilitate this transition, we present the first public, multi-label sewer defect classification dataset called Sewer-ML.

Sewer-ML consists of 1.3 million images of a large variety of sewer pipes annotated by professional sewer inspectors. The data is acquired from 75,618 inspection videos conducted over nine years. 12 methods from the sewer defect classification and multi-label classification domains are compared on Sewer-ML. Methods are evaluated using a novel, class-importance weighted F2 score, $F2_{\text{CIW}}$, which incorporates the economic impact of each class, and the F1 score for pipes with no annotated classes, $F1_{\text{Normal}}$. We present a benchmark algorithm by combining the best two-stage approach from the sewer domain with the best classifier from the multi-label domain, achieving a state-of-the-art performance with an $F2_{\text{CIW}}$ of 55.11% and $F1_{\text{Normal}}$ of 90.94%. The code, data, and trained models are open-sourced in order to lower the barrier of entry and encourage further development within sewer defect classification.

Acknowledgments. This research was funded by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project.

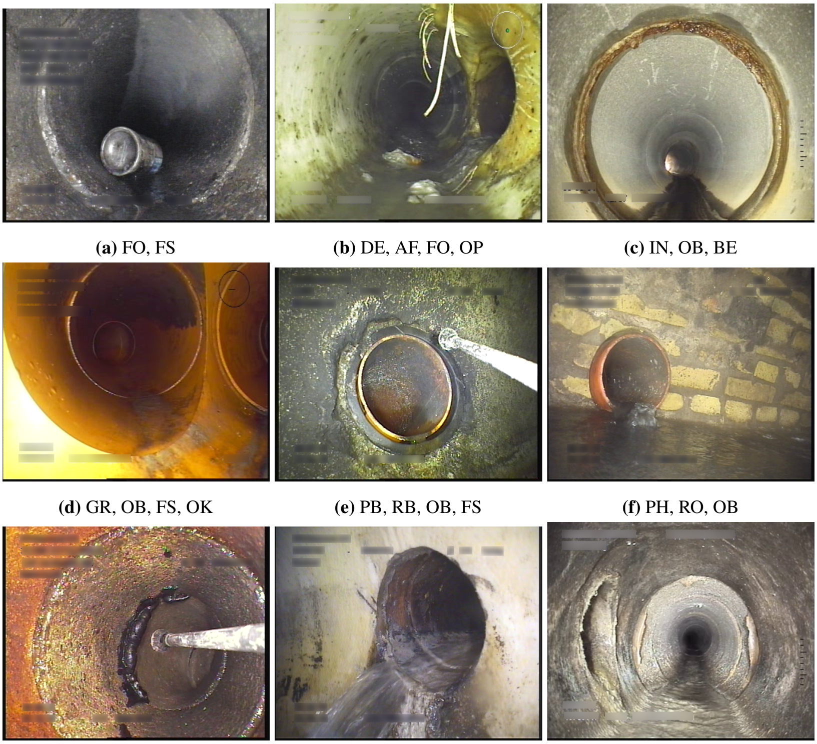
B.A Supplementary Materials Content

In these supplementary materials we describe in further detail aspects of the dataset and the training process and performance of the tested methods. Specifically, the following will be described:

- Additional examples from the Sewer-ML dataset (Section B.B).
- Further insights into the Sewer-ML dataset (Section B.C).
- Full training details and metric performance for the Faster-RCNN text detector (Section B.D).
- Full details on the Extra Trees hyperparameter grid search (Section B.E).
- The loss curves of the trained multi-label classification methods (Section B.F).
- Ablation study of the two-stage methods (Section B.G).
- Results when evaluating using the common multi-label performance metrics (Section B.H).

B.B Sewer-ML Dataset Examples

In this section we present more examples of the images in the Sewer-ML dataset. All images are annotated using the Danish inspection standard containing 18 classes [78], listed in Table B.1. In Figure B.5 we present examples of different cases with several co-occurring classes. In Figure B.6 we present five examples of each class, where only the mentioned class is present.



(a) FO, FS

(b) DE, AF, FO, OP

(c) IN, OB, BE

(d) GR, OB, FS, OK

(e) PB, RB, OB, FS

(f) PH, RO, OB

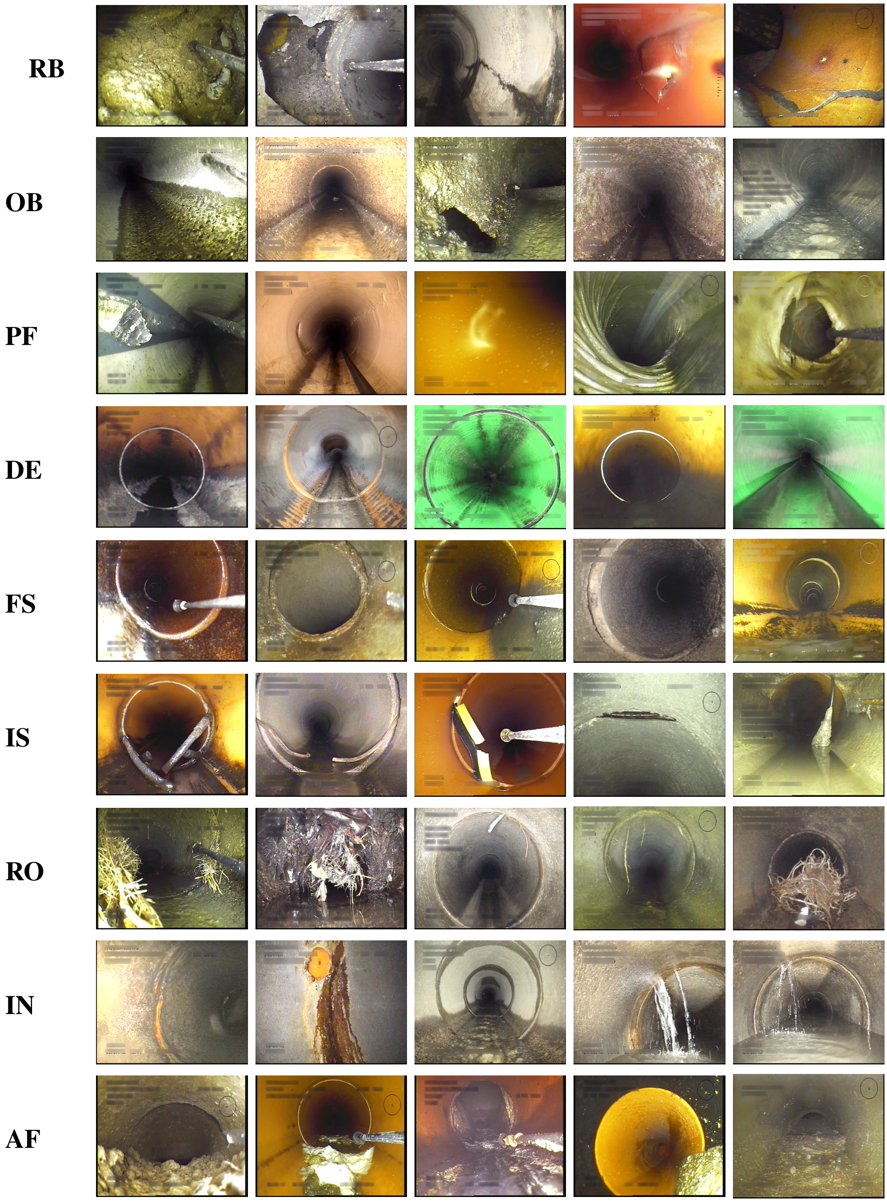
(g) IS, OK

(h) OS, PF

(i) OP, OK, FS, OB

Fig. B.5: Sewer-ML data examples with co-occurring classes. A subset of the images in the Sewer-ML showcasing images with multiple classes co-occurring and all annotated classes represented. The class codes are described in Table B.1.

Figure B.6: Sewer-ML data examples. A subset of the images in the Sewer-ML showcasing five images from each of the annotated classes as well as normal pipes in each row. The class codes are described in Table B.1.



Continued on next page

Figure B.6: **Continued from previous page**

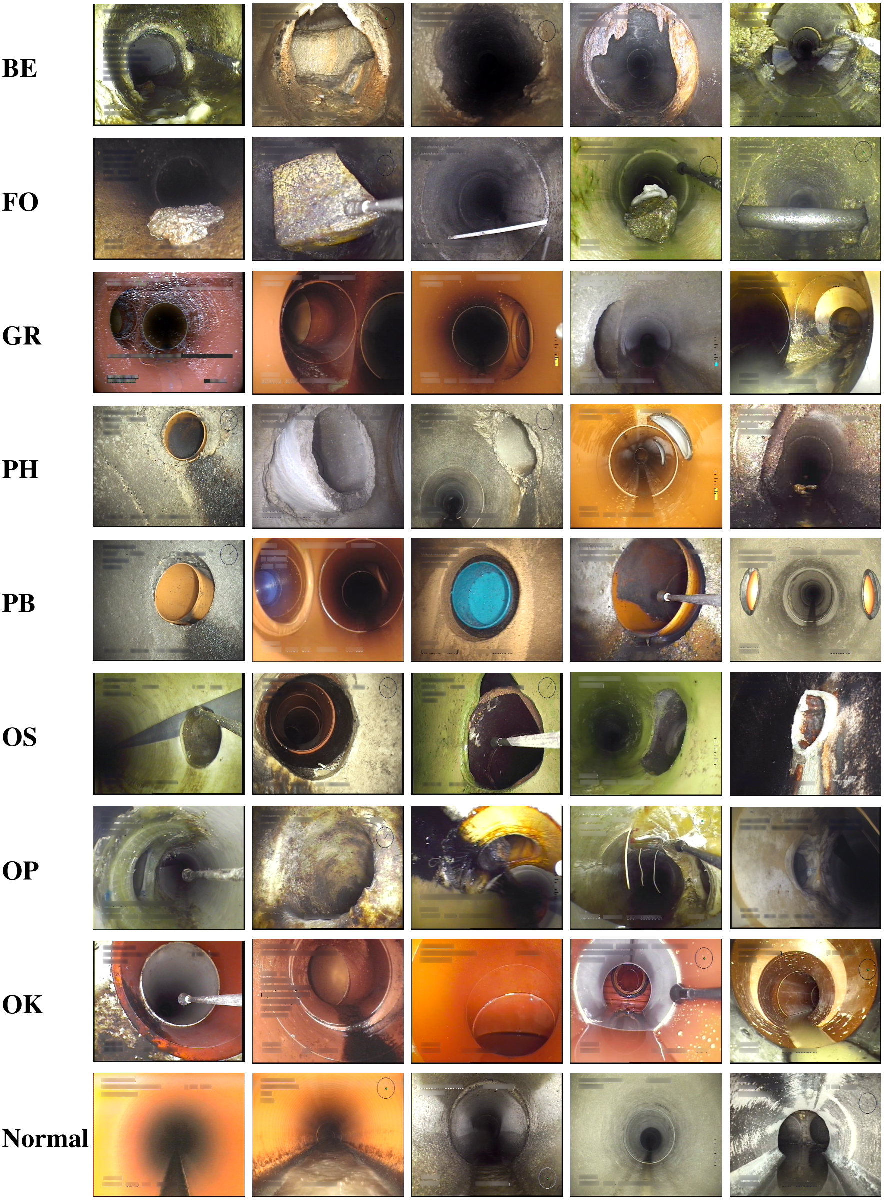


Table B.6: Class occurrences per split. The number of occurrences for each class per dataset split.

Split	RB	OB	PF	DE	FS	IS	RO	IN	AF
Training	45,821	184,379	16,254	19,084	283,983	6,271	22,637	23,782	74,856
Validation	5,538	23,624	2,021	2,038	36,218	881	2,917	2,812	9,059
Test	5,501	23,264	1,949	2,307	35,781	924	2,684	3,235	9,182
Total	56,860	231,267	20,224	23,429	355,982	8,076	28,238	29,829	93,097

Split	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Training	66,499	5,010	53,986	23,685	6,746	4,625	5,325	154,624	552,820
Validation	7,929	597	6,889	3,432	765	457	612	19,655	68,681
Test	8,720	649	6,726	2,962	833	530	533	19,420	69,221
Total	83,148	6,256	67,601	30,079	8,344	5,612	6,470	193,699	690,722

B.C Sewer-ML Dataset Insights

In this section, we describe the available information in the Sewer-ML dataset in more detail. First, we report the number of occurrences for each class in the dataset splits, see Table B.6, where it is observed that the distribution of the classes is similar across the different splits.

Moreover, we look into the pipe properties associated with each image. Each image contains information on the pipe shape, material, dimension, and water level.

In Figure B.7 we plot the distribution of the eight different pipe material types for the images in each split. We find that the concrete, vitrified clay, plastic, and lining materials are the most common materials in the Sewer-ML dataset. We also observe that all material types are equally represented across the splits, except for the “Brickwork” and “Unknown” material types. The reason these material types are skewed for the validation and test sets, is due to these materials being rarely used anymore, and therefore rarely occur in the sewer inspection videos. Therefore, the images containing these material types are from a small subset of pipes, which were not evenly spread out across the splits.

In Figure B.8 we plot the distribution of the six different pipe shapes for the images in each of the dataset splits. We find that the circular type is by far the most common pipe shape, followed secondly by conical pipes, whereas the remaining pipe shapes only appear a few thousand times each. As with the pipe material, we see that distribution of pipe shapes are similar between dataset splits, except for the “Eye shaped”, “Rectangular”, and “Other” pipe shapes. This is again due to these pipe shapes occurring in a limited set of sewer inspections, and have therefore not been evenly divided across the splits.

In Figure B.9 we plot the occurrences of the pipe dimensions associated with each image. The dimension is denoted in millimeters, as per the industry standard. We see that the majority of images are from pipes with a diameter of 100–1,000 millimeters,

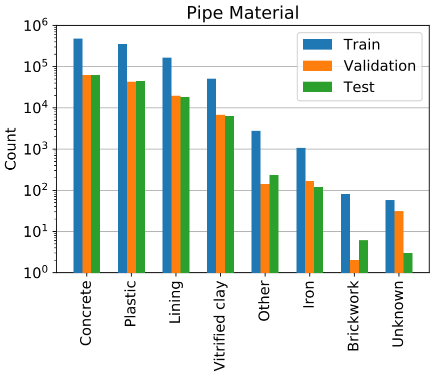


Fig. B.7: Distribution of the pipe materials. We plot the occurrence frequencies for each of the eight pipe materials in the dataset, for each dataset split. Note that the y-axis is log-scaled.

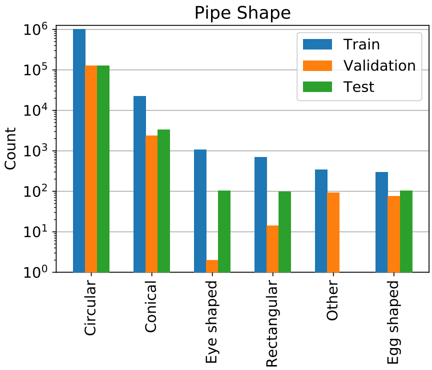


Fig. B.8: Distribution of the pipe shapes. We plot the occurrence frequencies for each of the six pipe shapes in the dataset, for each dataset split. Note that the y-axis is log-scaled.

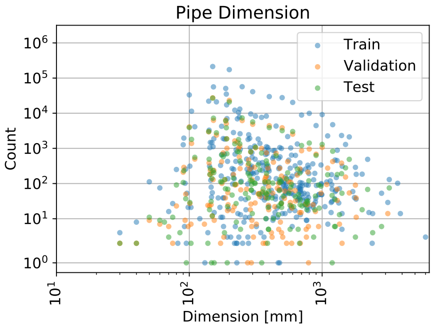


Fig. B.9: Distribution of the pipe dimensions. Plots of the occurrence frequencies of each pipe dimension, for each dataset split. Note that both axes are log-scaled.

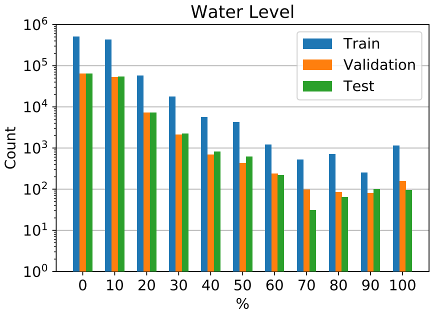


Fig. B.10: Distribution of the water level. We plot the occurrence frequencies for each of the water level classes, for each dataset split. Note that the y-axis is log-scaled.

with a skew towards 100 millimeters. We observe that the distribution of the pipe dimension for the training, validation, and test splits appears to be similar in shape, as expected.

In Figure B.10 we plot the distribution of the different water level classes for each data split. We find that the distribution of the water level classes is similar across the three dataset splits We also observe that the majority of the images have an associated water level in the range 0–30 %, while the remaining classes occur less often, and not as evenly split between the classes. This can be explained by the fact that when the majority of a pipe is filled with water, the inspections may at times be postponed for a later time and it becomes difficult to accurately access how much water it actually contains. Furthermore, the inspection vehicle will at times be partially or

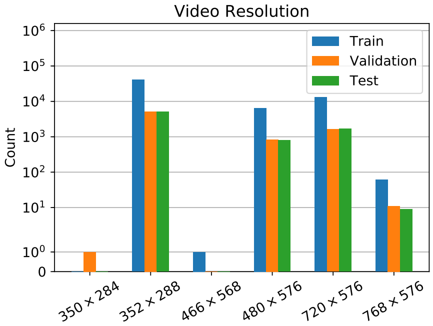


Fig. B.11: Distribution of video resolution. We present the distribution of the different resolutions for the videos in each dataset split. Note the y-axis is log-scaled.

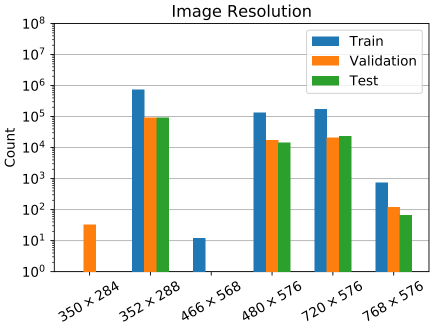


Fig. B.12: Distribution of image resolution. We present the distribution of the different resolutions for the images in each dataset split. Note the y-axis is log-scaled.

fully submerged in the water, resulting in the inspector losing key reference points used for estimating the water level, such as the pipe wall.

Lastly, in Figure B.11 we plot the resolution of the sewer inspection videos in each split. The resolution is denoted as width by height. It should be noted that the video resolutions reported are not the resolutions observed by the inspector. The videos are encoded in such a way that the video data is stored in the resolution reported in this work, but when presented using a media player the width is multiplied by a “sample aspect ratio”. We decide not to apply this resizing, in order to not introduce artifacts in the image data. We find that across the videos in each dataset split, the resolutions are evenly distributed. This is also true when looking at the resolution for all the images in the dataset splits, see Figure B.12.

B.D Faster-RCNN Training and Metric Details

In this section we detail the hyperparameters and training settings for the Faster-RCNN [82] model we use to redact overlaid text information on the images. We also present the full COCO [34] metric suite performance, to show how well the network performs. A training split of 20,739 images and a validation split of 2,305 images are used, wherein all text information is manually annotated with bounding boxes.

Hyperarameters. The Faster-RCNN model is trained for 26 epochs with a batch size of 16 batches. An SGD optimizer with momentum is used, with a learning rate of 0.02, momentum of 0.9 and weight decay of 0.0001. The learning rate is multiplied by 0.1 at epoch 16 and 22, respectively. We employ linear warm up of the learning rate during the first 1,000 mini batches of the first epoch, increasing the learning rate from 10^{-3} to 0.02. The backbone is a ResNet-50 FPN [29, 83] pre-trained on ImageNet [35], of which we fine-tune the last three residual blocks. Custom anchor boxes are used, with a bounding box ratios (height over width) of 1:8, 1:4 and 1:2, and bounding box

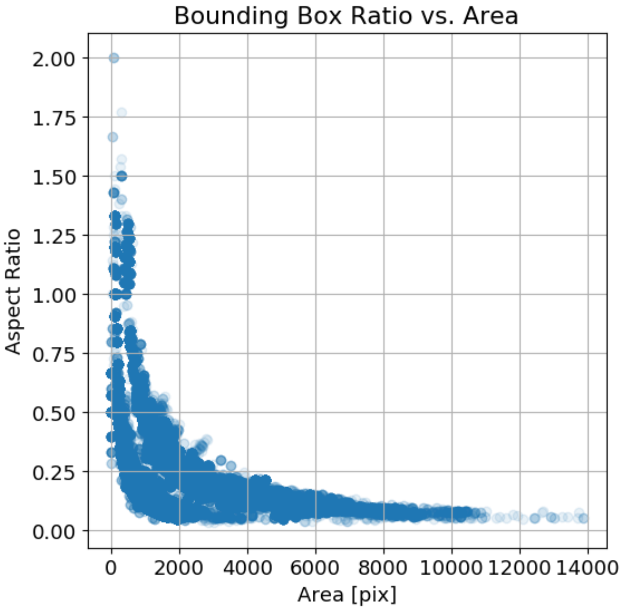


Fig. B.13: Training split bounding box information. The training split bounding box annotations are plotted with the bounding box area against the bounding box ratio.

Table B.7: Full COCO metric suite. The performance of the trained Faster-RCNN model on the validation set, for different Average Precision (AP) and Average Recall (AR) settings.

AP, IoU:			**AP@[0.5:0.95], Area:**			**AR@[0.5:0.95], #Dets:**			**AR@[0.5:0.95], Area:**		
0.5:0.95	0.5	0.75	S	M	L	1	10	100	S	M	L
89.10	98.89	96.39	88.08	89.96	95.63	10.06	88.31	92.25	91.72	92.71	96.28

scales with areas of 32^2, 64^2, 128^2, 256^2, and 512^2. These values are determined based on the bounding box information in the training split, see Figure B.13. All images are normalized using the ImageNet per channel mean and standard deviation, and horizontal flipping with a 50% chance is used during training. The images are rescaled such that the shortest side is 800 pixels, while enforcing that the largest side is no larger than 1,333 pixels. The training loss and mAP[0.5:0.95] on the validation set are plotted in Figure B.14.

Metrics. In order to determine the effect of the Faster-RCNN model, we compute the full COCO metrics suite on the validation set, as shown in Table B.7. As shown in the metrics, we have a high precision and recall, though the recall indicates that not all of the text objects have been detected. This is partially due to some text information being annotated with a single bounding box but detected as several boxes, and vice versa. To verify the annotations we manually inspect a set of randomly selected samples.

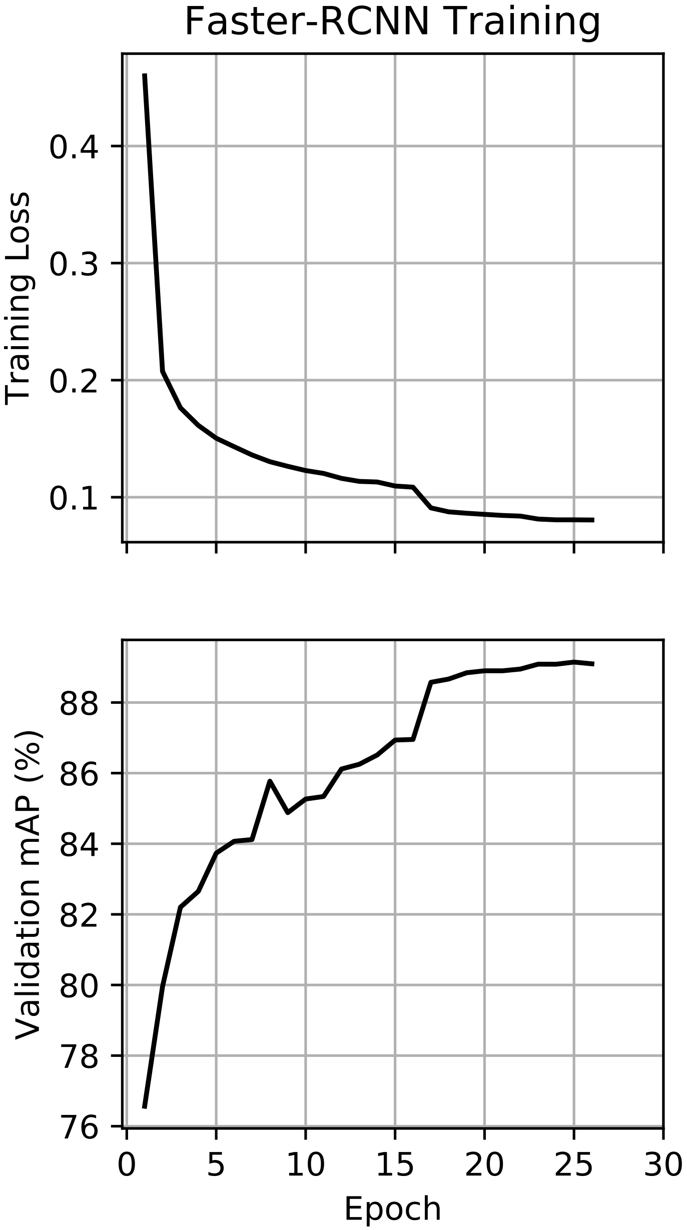


Fig. B.14: Faster-RCNN loss and metric curves. The training loss and validation metrics for the trained Faster-RCNN model. mAP@[0.5:0.95] is denoted as mAP.

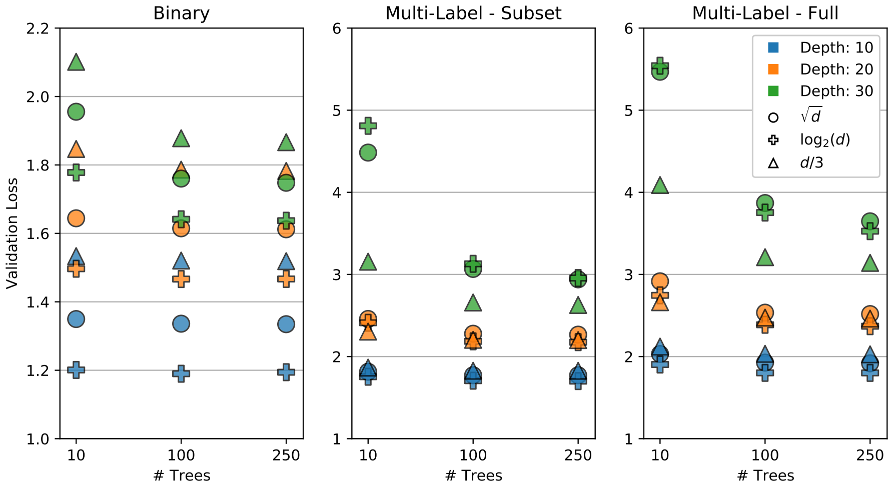


Fig. B.15: Extra Trees grid search results. Results of the grid search of the Extra Trees classifiers for: Binary classifier trained on full dataset, multi-label classifier trained on a subset of the dataset, and multi-label classifier trained on the full dataset.

B.E Extra Trees Hyperparameter Grid Search

For the system proposed by Myrans *et al.* [52, 53], two Extra Trees classifiers are used in sequence. However, the hyperparameters of the trees are not specified. Therefore, we conduct a small grid search across three hyperparameters: The amount of trees in the ensemble, the maximum depth of the trees, and the maximum amount of features used when splitting an internal node. The investigated parameters are reported in Table B.8. We train the Extra Trees classifier in three settings. First, we train under a binary setting determining whether there is *any* class in the image. Thereafter, we train a multi-label setting, first on a subset of the dataset only containing images with annotated classes, and secondly on the full dataset. The resulting validation losses of the hyperparameter search is shown in Figure B.15. From this we conclude that for the binary Extra Trees classifier 100 trees, with a maximum depth of 10 and using $\log_2(d)$ features when splitting, should be used. Similarly, we find that for the multi-label Extra Trees classifiers 250 trees, with a maximum depth of 10 and using $\log_2(d)$ features when splitting, should be used.

B.F CNN Loss Curves

We present the loss curves for all the tested convolutional neural networks (CNNs) tested, see Figure B.16. All networks are trained using the weighted binary cross-entropy loss, and using hyperparameters set based on the guidelines from Goyal *et*

Table B.8: Extra Trees grid search intervals. Hyperparameter search intervals for the Extra Trees classifiers. d denotes the dimensionality of the GIST descriptor.

Parameter	Values
Number of Trees	[10, 100, 250]
Max Depth	[10, 20, 30]
Max Features	$[\sqrt{d}, \log_2(d), d/3]$

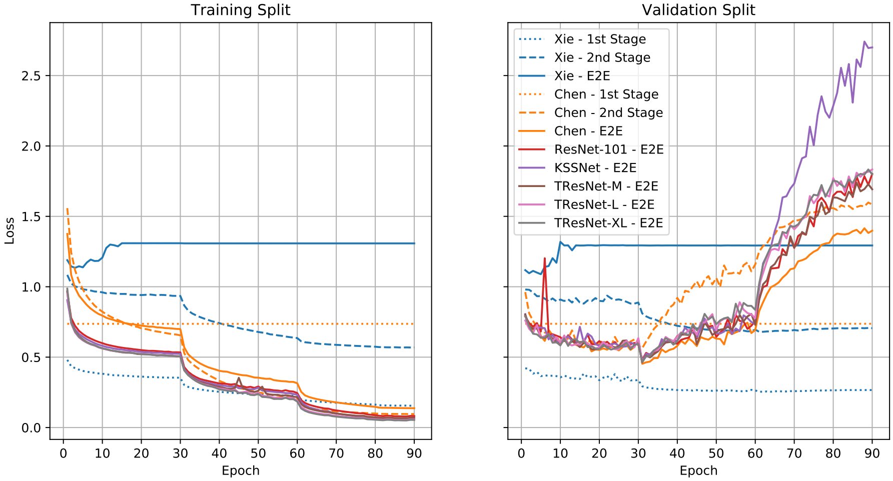


Fig. B.16: Multi-label CNN loss curves. The training and validation loss curves for all tested networks. “1st Stage” indicates a binary classifier, “2nd Stage” indicates a multi-label classifier trained on a subset of the dataset, and “E2E” indicates a multi-label classifier trained in an end-to-end manner with the full dataset.

al. [85]. Further training details are presented in the main manuscript.

From the loss plots we observe that the validation loss of the majority of the tested networks start diverging after approximately 30-40 epochs, a clear sign of overfitting. The method by Xie *et al.* [72] is an exception, with the first and second stage methods stagnating after 60–70 epochs. We also observe that the first stage of Chen *et al.* [67], the SqueezeNet [74], has a constant loss value for both the training and validation loss. Similarly, the second stage of Xie *et al.* settles on a constant loss after the initial 10 epochs when trained on the full dataset.

B.G Two-Stage Ablation Study

We conduct two ablation studies on the two-stage classifiers, to determine the effect of the different stages and training methodology.

What is the effect of the binary classifier? We compare the effect on performance of using both stages or only the second stage. These results are presented in Table B.9,

Table B.9: Effect of binary stage in two-stage classifiers. We present the metric performance for the two-stage methods, comparing the effect of the full pipeline and using only the multi-label classifier. TS denotes that both stages are used, otherwise only the second stage is used.

Model	TS	Validation		Test	
		$F2_{CIW}$ ↑	$F1_{Normal}$ ↑	$F2_{CIW}$ ↑	$F1_{Normal}$ ↑
Xie [72]	✓	**48.57**	**91.08**	**48.34**	**90.62**
		37.65	0.52	37.83	0.68
Chen [67]	✓	42.03	3.96	41.74	3.59
		42.03	3.96	41.74	3.59
Myrans [52]	✓	4.01	26.03	4.11	27.48
		19.25	0.00	19.19	0.00

Table B.10: Effect of training second stage on full dataset. The metric performance for the two-stage methods, when training both stages on the full dataset. TS denotes that both stages are used, otherwise only the second stage is used.

Model	TS	Validation		Test	
		$F2_{CIW}$ ↑	$F1_{Normal}$ ↑	$F2_{CIW}$ ↑	$F1_{Normal}$ ↑
Xie [72]	✓	31.98	**88.23**	31.82	**87.95**
		28.12	59.98	27.96	59.99
Chen [67]	✓	**43.45**	76.73	**43.14**	75.68
		43.45	76.73	**43.14**	75.68
Myrans [52]	✓	2.58	25.98	2.61	27.48
		7.48	0.00	7.37	0.00

and indicate that the first stage is crucial. Performance for Xie *et al.* [72] degrades for both metrics when the first stage is missing, whereas for Chen *et al.* [67] there is no difference as the first stage never predicts a normal pipe. For Myrans *et al.* [52] the first stage inaccurately classifies images with classes as normal pipes, causing a lower $F2_{CIW}$ score. This is improved when using only the second stage, but at the cost of an inability to recognize any normal pipes.

Training the second stage on the full dataset. Classically within the sewer classification domain, the second stage is only trained on data which contains some kind of class. We investigate whether performance improves by training on the full dataset, such that the second stage also sees normal pipes. The results are shown in Table B.10. For Myrans *et al.* the performance is reduced substantially in both tested settings, and the second stage is still unable to classify normal pipes. For Xie *et al.* both metrics are lower when comparing to Table B.9, except for the large increase in F_{Normal} score when only using the second stage. The only performance improvement is achieved by Chen *et al.* through the use of the deeper InceptionV3 network.

B.H Multi-Label Metrics and Results

When evaluating multi-label tasks, a large suite of metrics are commonly used, in order to uncover different aspects of the tested methods. Commonly, the F1-score is used in different variations, depending on how the F1-score is calculated or averaged. An overview of the different metrics is provided in Table B.11. Each of the metrics are in the range $[0,1]$, and for all a high score is better. As a reference on how to compute the metrics, we refer to the supplementary materials of the work by Durand *et al*. [24]. We present the classic performance metrics for each of the tested methods on both the validation and test splits, as well as the per-class F1, F2, Recall, Precision, and Average Precision (AP). It should be noted, that AP cannot be calculated for the normal class. This is due to the normal class being an implicit class, and therefore not possible to rank as it does not have a single associated probability. The Kumar *et al*. [69], Meijer *et al*. [71] and ML-GCN [23] methods are not shown in the metric tables as the models diverged during training. The benchmark algorithm consisting of the first stage from Xie *et al*. [72] and the TResNet-L multi-label classifier [31] is reported as "Benchmark". The metrics for the validation split are presented in Table B.12-B.17 and the metrics for the test split are presented in Table B.18-B.23.

Table B.11: Multi-label classification metrics. A short description of the commonly used multi-label classification metrics. For details on how the metrics are computed, we refer to Durand *et al.* [24].

Metric	**Description**
Macro-F1 (M-F1)	Average F1-score across all classes.
Micro-F1 (m-F1)	F1 score calculated over all samples.
Overall Precision (OV-P)	Precision metric calculated over all samples, regardless of class.
Overall Recall (OV-R)	Recall metric calculated over all samples, regardless of class.
Overall F1 (OV-F1)	F1 score calculated using OV-P and OV-R.
Per-class Precision (PC-P)	Average precision metric across all classes.
Per-class Recall (PC-R)	Average recall metric across all classes.
Per-class F1 (PC-F1)	F1-score calculated using PC-P and PC-R.
Zero-one Exact Match Accuracy (0-1)	Ratio of samples with all labels correctly predicted.
mean Average Precision (mAP)	Average of the Average Precision of all annotated classes

Table B.12: Performance metrics for each method - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	m-F1	M-F1	OV-F1	OV-P	OV-R	PC-F1	PC-P	PC-R	0-1	mAP
Sewer	Xie *et al.* [72]	59.33	38.10	59.33	46.31	82.52	42.43	29.61	74.79	51.64	66.40
	Chen *et al.* [67]	33.94	26.62	33.94	26.38	47.60	35.09	23.97	65.40	7.96	62.06
	Hassan *et al.* [68]	12.76	6.36	12.76	7.44	44.86	6.92	3.72	50.00	0.00	8.89
	Myrans *et al.* [52]	5.39	3.69	5.39	3.19	17.27	4.80	2.90	14.06	13.66	0.54
General	ResNet-101 [29]	54.47	38.08	54.47	40.63	82.62	43.58	28.98	87.83	39.96	76.27
	KSSNet [26]	56.18	39.37	56.18	42.52	82.77	44.82	30.12	87.56	41.28	77.63
	TResNet-M [31]	55.27	38.69	55.27	41.22	83.88	44.14	29.35	**88.93**	41.07	78.29
	TResNet-L [31]	56.01	39.63	56.01	42.09	83.69	44.90	30.10	88.32	41.22	78.75
	TResNet-XL [31]	55.83	39.30	55.83	41.82	83.98	44.66	29.85	88.67	41.68	78.32
	Benchmark	**61.45**	**42.39**	**61.45**	**47.02**	**88.67**	**46.38**	**32.25**	82.55	**51.65**	**79.79**

Table B.13: Per-class F1 score - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	22.97	71.40	35.83	22.84	77.12	9.77	17.94	28.39	40.27	38.09	**5.84**	49.03	37.86	24.21	13.80	29.03	70.29	91.08
	Chen *et al.* [67]	24.60	57.19	17.17	10.08	68.37	6.03	**31.78**	21.05	26.71	39.01	5.33	25.26	34.27	8.79	10.10	32.34	57.18	3.96
	Hassan *et al.* [68]	0.00	30.75	3.06	3.09	43.57	1.35	4.39	0.00	13.02	0.00	0.00	10.06	5.14	0.00	0.00	0.00	0.00	0.00
	Myrans *et al.* [52]	1.61	5.70	1.79	1.07	8.07	0.28	0.53	0.84	3.29	2.56	0.20	3.07	0.94	0.94	0.28	0.14	9.16	26.03
General	ResNet-101 [29]	24.08	73.10	30.46	18.47	79.44	9.48	20.43	27.80	39.92	41.07	4.50	47.41	40.62	24.66	18.08	32.83	73.52	79.55
	KSSNet [26]	**25.40**	73.64	31.52	18.84	80.59	10.56	21.06	28.88	41.11	42.15	4.82	51.24	43.69	25.23	19.28	35.55	74.58	80.60
	TResNet-M [31]	24.87	72.90	30.24	19.72	80.13	10.47	20.16	28.31	40.61	40.32	4.54	48.04	44.08	24.17	17.26	35.64	73.80	81.23
	TResNet-L [31]	24.51	73.14	30.87	19.74	79.94	11.57	19.76	29.49	41.24	41.49	4.74	50.31	47.15	28.00	17.88	38.95	73.34	81.22
	TResNet-XL [31]	24.75	73.15	32.20	20.58	79.89	10.21	19.76	29.09	40.30	41.15	4.69	48.35	45.22	26.45	18.23	37.08	74.57	81.81
	Benchmark	24.81	**74.50**	**36.39**	**23.91**	**80.69**	**11.87**	20.05	**31.19**	**42.39**	**42.63**	4.94	**51.40**	**48.53**	**33.09**	**21.58**	**48.75**	**75.00**	**91.32**

Table B.14: Per-class F2 score - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	38.87	77.86	52.92	37.48	79.78	19.96	32.90	46.59	53.52	53.48	**12.66**	58.74	56.49	40.44	26.37	47.51	76.98	90.32
	Chen *et al.* [67]	38.82	66.38	31.00	20.81	73.90	12.77	**43.33**	33.70	42.02	45.98	11.64	43.15	52.73	19.18	20.63	51.75	66.21	2.52
	Hassan *et al.* [68]	0.00	52.60	7.32	7.37	65.87	3.30	10.29	0.00	27.24	0.00	0.00	21.86	11.94	0.00	0.00	0.00	0.00	0.0
	Myrans *et al.* [52]	3.23	7.81	4.06	1.84	9.59	0.66	1.16	1.85	5.95	4.78	0.49	5.91	2.03	2.26	0.69	0.33	13.32	25.93
General	ResNet-101 [29]	42.45	84.34	51.08	35.34	87.49	19.98	37.81	47.47	59.18	59.87	10.39	64.78	61.24	44.03	34.81	54.23	82.99	71.60
	KSSNet [26]	**43.74**	**84.64**	52.24	35.92	87.45	21.76	38.67	48.54	59.89	**60.81**	11.08	**67.40**	63.94	44.73	36.60	57.05	83.30	72.95
	TResNet-M [31]	43.55	84.49	51.02	37.23	87.79	21.75	37.57	47.93	60.01	59.99	10.51	65.61	64.43	43.62	33.70	57.04	**83.71**	73.71
	TResNet-L [31]	43.08	84.39	51.75	37.39	**87.81**	23.50	37.03	49.20	**60.10**	60.60	10.91	67.05	**66.64**	48.24	34.53	60.12	83.59	73.69
	TResNet-XL [31]	43.34	84.44	52.99	38.50	87.69	21.37	37.01	48.93	59.79	60.42	10.79	65.77	65.07	46.46	35.12	58.61	83.63	74.49
	Benchmark	42.92	83.56	**54.06**	**39.16**	86.99	**23.89**	37.17	**50.41**	59.64	60.12	11.30	66.39	66.40	**50.16**	**37.32**	**64.86**	82.88	**90.79**

Table B.15: Per-class Precision score - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	13.66	62.73	23.30	13.83	**73.05**	5.28	10.21	17.20	28.51	25.75	**3.08**	**38.45**	24.43	14.51	7.69	17.61	61.39	92.36
	Chen *et al.* [67]	**15.27**	46.48	9.85	5.42	60.78	3.20	**22.00**	12.94	16.62	**31.15**	2.80	14.93	21.64	4.62	5.46	19.90	46.59	91.88
	Hassan *et al.* [68]	0.00	18.17	1.55	1.57	27.85	0.68	2.24	0.00	6.97	0.00	0.00	5.30	2.64	0.00	0.00	0.00	0.00	0.00
	Myrans *et al.* [52]	0.88	3.93	0.93	0.63	6.38	0.14	0.28	0.44	1.88	1.44	0.10	1.71	0.49	0.47	0.14	0.07	6.03	26.21
General	ResNet-101 [29]	13.99	59.82	18.21	10.29	68.89	5.05	11.56	16.44	25.88	26.96	2.32	32.76	26.02	14.23	10.04	19.80	61.78	97.61
	KSSNet [26]	14.95	60.52	18.98	10.51	71.26	5.69	11.97	17.24	26.99	27.89	2.48	36.61	28.59	14.61	10.78	21.84	63.50	97.68
	TResNet-M [31]	14.50	59.34	18.01	11.06	69.95	5.62	11.37	16.83	26.39	26.07	2.33	33.21	28.88	13.86	9.52	21.93	61.64	97.87
	TResNet-L [31]	14.26	59.84	18.46	11.05	69.55	6.27	11.12	17.69	27.08	27.19	2.44	35.53	31.70	16.48	9.91	24.54	60.88	**97.89**
	TResNet-XL [31]	14.43	59.81	19.47	11.59	69.58	5.46	11.12	17.36	26.12	26.87	2.41	33.55	29.98	15.40	10.12	23.00	63.16	97.86
	Benchmark	14.56	**63.11**	**23.56**	**14.50**	72.00	**6.46**	11.35	**19.07**	**28.60**	28.71	2.55	37.34	**33.50**	**21.11**	**12.67**	**34.49**	**64.74**	92.21

Table B.16: Per-class Recall score - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	72.16	82.86	77.59	65.46	81.67	65.49	74.08	81.33	68.54	73.19	57.29	67.67	84.06	73.07	67.18	82.52	82.20	89.83
	Chen *et al.* [67]	63.16	74.34	66.95	71.59	78.12	50.28	57.18	56.26	68.01	52.19	54.94	81.80	82.28	90.20	67.61	86.27	74.01	2.03
	Hassan *et al.* [68]	0.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	0.00	**100.00**	0.00	0.00	**100.00**	**100.00**	0.00	0.00	0.00	0.00	0.00
	Myrans *et al.* [52]	9.84	10.37	26.37	3.58	10.98	9.88	5.90	9.67	12.94	11.33	9.05	15.42	8.97	38.56	19.47	5.72	19.09	25.86
General	ResNet-101 [29]	86.38	93.96	93.07	90.28	93.82	76.50	87.42	**89.83**	87.25	86.16	80.90	85.73	92.57	92.42	90.81	**95.92**	90.78	67.13
	KSSNet [26]	84.36	94.01	92.97	90.73	92.72	74.23	87.38	88.90	86.12	86.25	82.41	85.34	92.54	92.29	91.25	95.59	90.34	68.61
	TResNet-M [31]	**87.25**	94.51	94.16	91.22	93.77	77.07	88.58	89.12	88.07	**88.93**	**85.26**	86.76	93.07	**94.12**	**92.34**	95.10	91.95	69.42
	TResNet-L [31]	87.04	94.04	94.26	92.59	93.98	75.14	88.79	88.73	86.46	87.45	83.08	86.17	91.99	93.07	91.03	94.28	**92.19**	69.39
	TResNet-XL [31]	86.85	94.13	93.02	91.81	93.80	78.89	88.58	89.69	88.22	87.84	82.24	86.56	91.99	93.73	91.90	95.59	91.00	70.29
	Benchmark	83.66	90.93	79.91	68.11	91.76	73.55	86.25	85.56	81.84	82.75	79.23	82.42	88.02	76.47	72.65	83.17	89.12	**90.44**

Table B.17: Per-class AP score - Validation Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK
Sewer	Xie *et al.* [72]	29.82	83.81	83.82	63.25	87.48	31.80	62.03	54.60	60.44	66.43	48.39	84.66	78.56	68.85	61.24	77.41	86.25
	Chen *et al.* [67]	48.30	73.98	49.98	45.03	83.03	56.31	78.12	45.85	61.99	71.01	42.94	74.62	78.27	56.95	51.00	56.89	80.80
	Hassan *et al.* [68]	7.20	32.00	1.09	0.26	37.38	1.26	5.90	6.25	7.71	11.05	1.44	10.27	4.34	0.00	0.00	5.85	19.14
	Myrans *et al.* [52]	0.05	0.62	0.10	0.64	2.57	0.13	0.12	0.19	1.02	0.28	0.00	0.35	0.00	1.17	0.00	0.00	1.96
General	ResNet-101 [29]	54.54	90.21	84.15	76.73	93.56	49.33	81.88	67.13	74.85	80.20	64.11	90.83	87.63	66.49	59.77	81.58	93.57
	KSSNet [26]	56.86	90.74	85.42	76.50	94.05	56.75	**83.43**	68.86	75.14	81.40	**65.51**	91.27	89.20	66.49	64.58	79.66	93.87
	TResNet-M [31]	**57.22**	90.90	87.74	77.69	93.98	58.68	80.56	**69.94**	**76.17**	**82.39**	60.67	91.55	**89.90**	69.27	65.58	84.39	94.32
	TResNet-L [31]	56.76	90.75	88.32	78.36	93.95	60.42	80.88	69.21	75.64	81.99	64.62	91.37	89.38	69.75	69.39	83.57	94.44
	TResNet-XL [31]	57.15	90.81	87.36	78.34	94.04	56.91	80.92	69.84	76.01	82.00	63.28	91.69	88.97	69.66	68.16	82.09	94.23
	Benchmark	56.68	**90.93**	**90.12**	**80.30**	**94.06**	**60.55**	80.79	69.45	75.99	82.27	65.32	**92.06**	89.89	**75.70**	**72.97**	**84.81**	**94.57**

Table B.18: Performance metrics for each method - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	m-F1	M-F1	OV-F1	OV-P	OV-R	PC-F1	PC-P	PC-R	0-1	mAP
Sewer	Xie *et al*. [72]	59.05	37.94	59.05	46.06	82.24	42.16	29.48	73.95	51.55	65.32
	Chen *et al*. [67]	33.49	26.23	33.49	26.03	46.94	34.55	23.60	64.48	7.63	59.89
	Hassan *et al*. [68]	12.57	6.27	12.57	7.33	44.12	6.83	3.67	50.00	0.00	7.35
	Myrans *et al*. [52]	5.66	3.88	5.66	3.36	18.07	5.02	3.04	14.43	14.51	0.59
General	ResNet-101 [29]	53.91	37.94	53.91	40.19	81.85	43.46	28.89	87.70	39.38	74.99
	KSSNet [26]	55.64	39.22	55.64	42.12	81.96	44.68	30.02	87.32	40.46	75.70
	TResNet-M [31]	54.62	38.53	54.62	40.72	82.94	43.96	29.24	**88.56**	40.23	76.55
	TResNet-L [31]	55.34	39.45	55.34	41.56	82.79	44.72	29.97	88.05	40.42	76.82
	TResNet-XL [31]	55.08	38.98	55.08	41.21	83.01	44.34	29.61	88.23	40.74	76.61
	Benchmark	**61.26**	**42.35**	**61.26**	**46.90**	**88.30**	**46.22**	**32.24**	81.61	**51.59**	**77.79**

Table B.19: Per-class F1 score - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al*. [72]	23.47	71.12	34.63	23.51	77.25	9.99	16.14	31.33	39.59	39.93	**6.29**	49.22	34.61	24.09	14.27	27.05	69.85	90.62
	Chen *et al*. [67]	24.54	57.24	16.85	11.65	67.47	6.09	**29.99**	23.03	26.47	37.67	5.62	24.84	31.53	8.58	11.01	29.58	56.46	3.59
	Hassan *et al*. [68]	0.00	30.35	2.95	3.49	43.16	1.41	4.04	0.00	13.19	0.00	0.00	9.84	4.45	0.00	0.00	0.00	0.00	0.00
	Myrans *et al*. [52]	1.63	5.60	1.71	1.46	8.18	0.36	0.60	1.02	4.03	2.93	0.21	3.16	0.82	0.89	0.31	0.16	9.28	27.48
General	ResNet-101 [29]	24.42	72.52	28.62	19.75	79.22	9.98	19.07	30.13	38.93	42.15	4.87	47.69	37.80	26.86	18.56	30.54	73.28	78.57
	KSSNet [26]	**26.06**	73.34	29.89	19.64	**80.56**	10.83	19.84	31.47	40.59	43.50	5.24	50.88	40.74	26.39	20.55	32.84	74.38	79.29
	TResNet-M [31]	24.78	72.54	28.94	20.96	79.87	10.89	18.52	31.14	39.64	41.39	4.90	47.97	40.11	24.99	18.34	34.68	73.90	79.91
	TResNet-L [31]	24.78	72.93	28.68	20.62	79.60	12.01	18.20	32.29	40.43	42.56	5.11	49.97	43.33	28.13	19.43	38.68	73.41	79.88
	TResNet-XL [31]	24.76	72.66	30.24	21.49	79.71	10.46	18.32	31.51	39.59	41.94	5.13	48.32	41.21	27.12	19.25	35.10	74.41	80.42
	Benchmark	25.11	**74.40**	**35.58**	**25.01**	80.50	**12.26**	18.59	**34.26**	**41.93**	**44.16**	5.26	**51.28**	**45.09**	**31.60**	**22.20**	**49.17**	**75.04**	**90.94**

Table B.20: Per-class F2 score - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	39.77	77.88	51.49	37.53	79.90	20.42	30.19	49.28	53.16	54.67	**13.50**	59.40	53.74	39.57	26.30	45.24	76.63	89.77
	Chen *et al.* [67]	38.91	66.52	29.90	23.33	73.08	12.94	**42.11**	35.50	41.72	44.01	12.11	42.63	50.53	18.73	21.87	48.37	65.92	2.28
	Hassan *et al.* [68]	0.00	52.14	7.07	8.28	65.50	3.45	9.53	0.00	27.53	0.00	0.00	21.43	10.44	0.00	0.00	0.00	0.00	0.0
	Myrans *et al.* [52]	3.27	7.70	3.89	2.47	9.75	0.87	1.33	2.22	7.27	5.35	0.51	6.11	1.80	2.15	0.77	0.38	13.51	27.35
General	ResNet-101 [29]	42.95	83.92	48.25	37.06	87.22	21.11	35.87	50.06	58.23	60.37	11.18	64.81	58.83	47.12	35.24	51.76	82.63	70.41
	KSSNet [26]	**44.79**	**84.52**	49.68	36.93	87.36	22.42	36.96	51.45	59.54	**61.55**	11.96	**66.99**	61.57	46.35	**37.92**	54.59	83.00	71.32
	TResNet-M [31]	43.39	84.36	48.99	38.84	**87.48**	22.70	35.12	51.09	59.13	60.27	11.27	65.41	60.99	44.71	35.04	56.55	**83.72**	72.08
	TResNet-L [31]	43.50	84.42	48.55	38.39	87.45	24.45	34.67	52.27	**59.55**	61.13	11.70	66.37	63.54	**48.58**	36.69	60.42	83.50	72.03
	TResNet-XL [31]	43.39	84.22	50.14	39.42	87.43	22.00	34.89	51.39	59.15	60.64	11.74	65.47	61.83	47.45	36.31	56.76	83.52	72.74
	Benchmark	43.35	83.82	**52.94**	**39.69**	86.76	**24.70**	34.96	**53.41**	59.45	61.05	11.94	66.05	**64.00**	47.39	36.79	**65.41**	82.72	**90.35**

Table B.21: Per-class Precision score - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	13.95	62.14	22.40	14.49	**73.20**	5.40	9.09	19.50	27.77	27.54	**3.33**	**38.28**	21.72	14.58	8.10	16.20	60.87	92.09
	Chen *et al.* [67]	15.19	46.43	9.75	6.35	59.82	3.24	**20.27**	14.52	16.45	**30.38**	2.97	14.65	19.38	4.51	6.02	17.96	45.57	91.35
	Hassan *et al.* [68]	0.00	17.89	1.50	1.77	27.52	0.71	2.06	0.00	7.06	0.00	0.00	5.17	2.28	0.00	0.00	0.00	0.00	0.00
	Myrans *et al.* [52]	0.89	3.86	0.88	0.87	6.46	0.18	0.31	0.54	2.32	1.67	0.11	1.75	0.43	0.45	0.16	0.08	6.09	27.70
General	ResNet-101 [29]	14.20	59.14	17.06	11.10	68.72	5.31	10.71	18.11	25.08	28.04	2.51	33.12	23.69	15.65	10.37	18.14	61.65	97.37
	KSSNet [26]	**15.36**	60.10	17.96	11.03	71.31	5.82	11.20	19.11	26.52	29.22	2.71	36.33	26.05	15.36	11.66	19.73	63.39	97.46
	TResNet-M [31]	14.45	58.81	17.21	11.86	69.76	5.83	10.36	18.86	25.58	27.19	2.52	33.21	25.54	14.40	10.23	21.09	61.81	97.59
	TResNet-L [31]	14.43	59.44	17.05	11.64	69.23	6.50	10.15	19.73	26.33	28.26	2.64	35.40	28.32	16.53	10.89	24.18	61.10	**97.62**
	TResNet-XL [31]	14.43	59.12	18.20	12.22	69.48	5.58	10.23	19.16	25.52	27.70	2.65	33.64	26.48	15.83	10.80	21.45	62.96	97.58
	Benchmark	14.76	**62.66**	**23.01**	**15.47**	71.87	**6.66**	10.44	**21.44**	**28.11**	30.22	2.72	37.36	**30.22**	**20.32**	**13.37**	**34.79**	**64.99**	91.94

Table B.22: Per-class Recall score - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK	Normal
Sewer	Xie *et al.* [72]	74.02	83.15	76.24	62.29	81.77	66.99	71.98	79.72	68.91	72.53	57.16	68.90	85.11	69.27	60.00	81.99	81.93	89.21
	Chen *et al.* [67]	63.81	74.59	61.83	70.35	77.37	51.41	57.64	55.55	67.74	49.56	52.39	81.56	84.47	88.48	63.96	83.86	74.21	1.83
	Hassan *et al.* [68]	0.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	0.00	**100.00**	0.00	0.00	**100.00**	**100.00**	0.00	0.00	0.00	0.00	0.00
	Myrans *et al.* [52]	9.96	10.26	26.37	4.55	11.17	12.34	7.19	10.26	15.62	11.87	8.63	16.13	8.98	33.61	18.49	7.69	19.42	27.26
General	ResNet-101 [29]	86.91	93.73	88.92	89.16	93.52	82.47	86.85	**89.55**	86.96	84.82	81.66	85.19	93.52	94.72	87.92	96.44	90.31	65.85
	KSSNet [26]	85.98	94.07	88.92	89.34	92.58	78.25	87.07	89.21	86.46	85.09	81.66	84.91	93.42	93.52	86.79	**97.75**	89.96	66.83
	TResNet-M [31]	86.95	94.64	91.02	90.03	93.42	82.03	87.22	89.21	87.95	**86.62**	**84.75**	86.34	93.38	94.36	89.06	97.56	91.85	67.66
	TResNet-L [31]	**87.64**	94.34	90.20	90.16	93.61	79.00	87.48	88.93	86.99	86.19	82.74	84.94	92.20	94.24	**90.00**	96.62	**91.93**	67.60
	TResNet-XL [31]	87.04	94.22	89.33	88.86	93.47	83.23	87.89	88.72	88.22	86.27	83.05	85.74	92.81	**94.84**	88.68	96.44	90.94	68.39
	Benchmark	84.04	91.54	78.45	65.19	91.50	76.52	84.72	85.16	82.42	81.95	77.81	81.74	88.83	71.07	65.47	83.86	88.78	**89.96**

Table B.23: Per-class AP score - Test Split. The metrics are presented as percentages, and the highest score in each column is denoted in bold.

	Model	RB	OB	PF	DE	FS	IS	RO	IN	AF	BE	FO	GR	PH	PB	OS	OP	OK
Sewer	Xie *et al.* [72]	35.07	83.48	85.86	62.97	87.30	36.64	59.04	58.52	58.15	62.44	36.26	82.11	80.02	65.49	53.70	77.47	85.85
	Chen *et al.* [67]	48.49	74.08	48.09	47.43	81.76	57.95	74.70	48.77	60.86	65.30	31.56	74.91	80.58	43.37	49.87	49.86	80.61
	Hassan *et al.* [68]	6.16	26.79	0.33	3.60	35.10	1.17	1.25	3.16	5.62	5.14	0.44	9.23	4.17	0.00	1.57	2.52	18.69
	Myrans *et al.* [52]	0.11	0.69	0.00	0.63	2.91	0.00	0.09	0.16	1.80	0.43	0.18	0.69	0.00	0.00	0.00	0.24	2.09
General	ResNet-101 [29]	55.25	90.20	88.04	71.96	93.32	65.63	78.98	65.69	71.40	77.78	47.33	90.72	88.05	65.34	52.55	79.33	93.32
	KSSNet [26]	**58.43**	**90.59**	86.80	71.99	93.68	69.97	**80.75**	68.28	72.57	78.92	44.03	91.11	88.46	62.31	55.86	79.26	93.83
	TResNet-M [31]	55.61	90.16	89.82	**76.00**	93.65	65.85	78.58	69.71	73.45	79.82	50.44	91.03	**89.41**	67.57	57.79	78.11	94.32
	TResNet-L [31]	56.95	90.38	89.28	75.03	93.64	68.61	80.04	70.09	73.59	79.43	**48.74**	91.36	88.77	67.29	59.38	79.00	94.40
	TResNet-XL [31]	56.64	90.00	89.17	75.66	93.68	63.87	79.70	68.90	73.62	79.80	48.14	91.33	88.83	69.51	59.38	79.96	94.17
	Benchmark	56.99	90.46	**89.89**	75.09	**93.70**	**70.74**	80.20	**70.81**	**73.99**	**79.96**	48.21	**92.21**	89.08	**72.70**	**62.57**	**81.33**	**94.55**

References

[1] American Society of Civil Engineers, “2017 infrastructure report card - wastewater,” 2017, accessed: 08-11-2020. [Online]. Available: https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf

[2] J. B. Haurum and T. B. Moeslund, “A survey on image-based automation of cctv and sset sewer inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[3] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[4] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, “Deep convolutional ranking for multilabel image annotation,” *arXiv:1312.4894*, 2013.

[5] A. Kanehira and T. Harada, “Multi-label ranking from positive and unlabeled data,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5138–5146.

[6] Y. Li, Y. Song, and J. Luo, “Improving pairwise ranking for multi-label image classification,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1837–1845.

[7] B.-B. Gao and H.-Y. Zhou, “Multi-label recognition with multi-class attentional regions,” *arXiv:2007.01755*, 2020.

[8] W. Ge, S. Yang, and Y. Yu, “Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1277–1286.

[9] H. Guo, K. Zhang, X. Fan, H. Yu, and S. Wang, “Visual attention consistency under image transforms for multi-label image classification,” in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 729–739.

[10] P. Li, P. Chen, Y. Xie, and D. Zhang, “Bi-modal learning with channel-wise attention for multi-label image classification,” *IEEE Access*, vol. 8, pp. 9965–9977, 2020.

[11] Y. Liu, L. Sheng, J. Shao, J. Yan, S. Xiang, and C. Pan, “Multi-label image classification via knowledge distillation from weakly-supervised detection,” in *ACM International Conference on Multimedia*, 2018, pp. 700–708.

[12] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, “Multi-label image recognition by recurrently discovering attentional regions,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 464–472.

[13] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “Hcp: A flexible cnn framework for multi-label image classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1901–1907, 2016.

[14] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, “Exploit bounding box annotations for multi-label object recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 280–288.

[15] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, “Multilabel image classification with regional latent semantic dependencies,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2801–2813, 2018.

[16] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, “Learning spatial regularization with image-level supervisions for multi-label image classification,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2027–2036.

[17] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, “Order-free rnn with visual attention for multi-label classification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[18] T. Chen, Z. Wang, G. Li, and L. Lin, “Recurrent attentional reinforcement learning for multi-label image recognition,” *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[19] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, “Cnn-rnn: A unified framework for multi-label image classification,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2285–2294.

[20] V. O. Yazici, A. Gonzalez-Garcia, A. Ramisa, B. Twardowski, and J. van de Weijer, “Orderless recurrent models for multi-label classification,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 13 437–13 446.

[21] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, “Learning semantic-specific graph representation for multi-label image recognition,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 522–531.

[22] Z.-M. Chen, X.-S. Wei, X. Jin, and Y. Guo, “Multi-label image recognition with joint class-aware map disentangling and label correlation embedding,” in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 622–627.

[23] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5172–5181.

[24] T. Durand, N. Mehrasa, and G. Mori, “Learning a deep convnet for multi-label classification with partial labels,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 647–657.

[25] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. F. Wang, “Multi-label zero-shot learning with structured knowledge graphs,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1576–1585.

[26] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, “Multi-label classification with label graph superimposing,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 265–12 272, Apr. 2020.

[27] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, “Cross-modality attention with semantic graph embedding for multi-label classification,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12 709–12 716, Apr. 2020.

[28] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, pp. 770–778.

[30] B. Wu, W. Chen, Y. Fan, Y. Zhang, J. Hou, J. Liu, and T. Zhang, “Tencent ml-images: A large-scale multi-label image database for visual representation learning,” *IEEE Access*, vol. 7, pp. 172 683–172 693, 2019.

[31] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, “Tresnet: High performance gpu-dedicated architecture,” *arXiv:2003.13630*, 2020.

[32] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[33] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, “Nus-wide: A real-world web image database from national university of singapore,” in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR’09)*, Santorini, Greece., July 8-10, 2009.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision*

– *ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[36] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, and V. Ferrari, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, Mar. 2020.

[37] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," *arXiv:2009.14119*, 2020.

[38] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007.

[40] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, "Distribution-balanced loss for multi-label classification in long-tailed datasets," in *European Conference on Computer Vision (ECCV)*, 2020.

[41] O. Duran, K. Althoefer, and L. D. Seneviratne, "State of the art in sensor technologies for sewer inspection," *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, Apr 2002.

[42] Z. Liu and Y. Kleiner, "State of the art review of inspection technologies for condition assessment of water pipes," *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

[43] S. Iyer, S. K. Sinha, M. K. Pedrick, and B. R. Tittmann, "Evaluation of ultrasonic inspection and imaging systems for concrete pipes," *Automation in Construction*, vol. 22, pp. 149 – 164, 2012, planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[44] M. S. Khan and R. Patil, "Acoustic characterization of pvc sewer pipes for crack detection using frequency domain analysis," in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–5.

[45] ——, "Statistical analysis of acoustic response of pvc pipes for crack detection," in *SoutheastCon 2018*, 2018, pp. 1–5.

[46] M. Lepot, N. Stanić, and F. H. Clemens, “A technology for sewer pipe inspection (part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification,” *Automation in Construction*, vol. 73, pp. 1 – 11, 2017.

[47] A. D. Tezerjani, M. Mehrandezh, and R. Paranjape, “Defect detection in pipes using a mobile laser-optics technology and digital geometry,” *MATEC Web of Conferences*, vol. 32, p. 06006, 2015.

[48] D. Alejo, F. Caballero, and L. Merino, “Rgbd-based robot localization in sewer networks,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4070–4076.

[49] D. Alejo, G. Mier, C. Marques, F. Caballero, L. Merino, and P. Alvito, *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*. Cham: Springer International Publishing, 2020, pp. 275–296.

[50] J. B. Haurum, M. M. J. Allahham., M. S. Lynge., K. S. Henriksen, I. A. Nikolov., and T. B. Moeslund., “Sewer defect classification using synthetic point clouds,” in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2021, pp. 891–900.

[51] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, “Generating synthetic point clouds of sewer networks: An initial investigation,” in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2020, pp. 364–373.

[52] J. Myrans, R. Everson, and Z. Kapelan, “Automated detection of fault types in cctv sewer surveys,” *Journal of Hydroinformatics*, vol. 21, no. 1, pp. 153–163, Oct 2018.

[53] ——, “Automated detection of faults in sewers using CCTV image sequences,” *Automation in Construction*, vol. 95, pp. 64–71, Nov. 2018.

[54] X. Ye, J. Zuo, R. Li, Y. Wang, L. Gan, Z. Yu, and X. Hu, “Diagnosis of sewer pipe defects on image recognition of multi-features and support vector machine in a southern chinese city,” *Frontiers of Environmental Science & Engineering*, vol. 13, no. 2, Jan. 2019.

[55] X. Fang, W. Guo, Q. Li, J. Zhu, Z. Chen, J. Yu, B. Zhou, and H. Yang, “Sewer pipeline fault identification using anomaly detection algorithms on video sequences,” *IEEE Access*, vol. 8, pp. 39 574–39 586, 2020.

[56] S. Moradi and T. Zayed, “Real-time defect detection in sewer closed circuit television inspection videos,” in *Pipelines 2017*, 2017, pp. 295–307.

[57] S. Moradi, T. Zayed, F. Nasiri, and F. Golkhoo, “Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition,” *Journal of Infrastructure Systems*, vol. 26, no. 3, p. 04020018, 2020.

[58] M. Wang, S. S. Kumar, and J. C. Cheng, “Automated sewer pipe defect tracking in cctv videos based on defect detection and metric learning,” *Automation in Construction*, vol. 121, p. 103438, 2021.

[59] J. B. Haurum, C. H. Bahnsen, M. Pedersen, and T. B. Moeslund, “Water level estimation in sewer pipes using deep convolutional neural networks,” *Water*, vol. 12, no. 12, 2020.

[60] H. Ji, S. Yoo, B.-J. Lee, D. Koo, and J.-H. Kang, “Measurement of wastewater discharge in sewer pipes using image analysis,” *Water*, vol. 12, no. 6, p. 1771, Jun 2020.

[61] J. C. Cheng and M. Wang, “Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,” *Automation in Construction*, vol. 95, pp. 155 – 171, 2018.

[62] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, “Deep learning-based automated detection of sewer defects in cctv videos,” *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[63] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, “A deep learning-based framework for an automated defect detection system for sewer pipes,” *Automation in Construction*, vol. 109, p. 102967, 2020.

[64] G. Pan, Y. Zheng, S. Guo, and Y. Lv, “Automatic sewer pipe defect semantic segmentation based on improved u-net,” *Automation in Construction*, vol. 119, p. 103383, 2020.

[65] C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, “A vision-based system for internal pipeline inspection,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3289–3299, 2019.

[66] M. Wang and J. C. P. Cheng, “A unified convolutional neural network integrated with conditional random field for pipe defect segmentation,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, pp. 162–177, 2020.

[67] K. Chen, H. Hu, C. Chen, L. Chen, and C. He, “An intelligent sewer defect detection method based on convolutional neural network,” in *2018 IEEE International Conference on Information and Automation (ICIA)*, Aug 2018, pp. 1301–1306.

[68] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, “Underground sewer pipe condition assessment based on convolutional neural networks,” *Automation in Construction*, vol. 106, p. 102849, 2019.

[69] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, “Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks,” *Automation in Construction*, vol. 91, pp. 273 – 283, 2018.

[70] D. Li, A. Cong, and S. Guo, “Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification,” *Automation in Construction*, vol. 101, pp. 199 – 208, 2019.

[71] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, “A defect classification methodology for sewer image sets with convolutional neural networks,” *Automation in Construction*, vol. 104, pp. 281 – 298, 2019.

[72] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, “Automatic detection and classification of sewer defects via hierarchical deep learning,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2019.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[74] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size,” *arXiv:1602.07360*, 2016.

[75] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[76] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, May 2001.

[77] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr 2006.

[78] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 6th ed. Dansk Vand og Spildevandsforening (DANVA), 2010.

[79] ——, *Fotomanualen: Beregning af Fysisk Indeks ved TV-inspektion*, 1st ed. Dansk Vand og Spildevandsforening (DANVA), 2005.

[80] G. Tsoumakas, I. Katakis, and I. Vlahavas, *Mining Multi-label Data*. Boston, MA: Springer US, 2010, pp. 667–685.

[81] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 71–79.

[82] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[83] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[84] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, Mar. 2021.

[85] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch sgd: Training imagenet in 1 hour," *arXiv:1706.02677*, 2017.

[86] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. USA: Butterworth-Heinemann, 1979.

[87] J. Atwood, P. Baljekar, P. Barnes, A. Batra, E. Breck, P. Chi, T. Doshi, J. Elliott, G. Kour, A. Gaur, Y. Halpern, H. Jicha, M. Long, J. Saxena, R. Singh, and D. Sculley, "Inclusive images challenge," 2018, accessed: 16-11-2020. [Online]. Available: https://www.kaggle.com/c/inclusive-images-challenge/overview/evaluation

[88] Planet, "Planet: Understanding the amazon from space," 2017, accessed: 16-11-2020. [Online]. Available: https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/overview/evaluation

[89] C. Zhang, G. Vesom, J. Choi, M. Kessler, and T. Loic, "imet collection 2019 - fgvc6," 2019, accessed: 16-11-2020. [Online]. Available: https://www.kaggle.com/c/imet-2019-fgvc6/overview/evaluation

Part III

Advancing Image-based Automation of Sewer Inspections

Paper C

Water Level Estimation in Sewer Pipes Using Deep Convolutional Neural Networks

Joakim Bruslund Haurum, Chris H. Bahnsen, Malte Pedersen and Thomas B. Moeslund

The paper has been published in the
Water, vol. 12, nbr. 12, p. 3412, 2020.

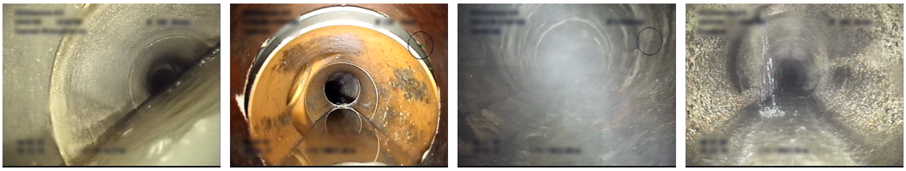
The layout has been revised.

Abstract

Sewer pipe inspections are currently conducted by professionals who remotely control a robot from above ground. This expensive and slow approach is prone to human mistakes. Therefore, there is both an economic and scientific interest in automating the inspection process by creating systems able to recognize sewer defects. However, the extent of research put into automatic water level estimation in sewers has been limited despite being a prerequisite for further analysis of the pipe as only sections above the water level can be visually inspected. In this work, we utilize a dataset of still images obtained from over 5000 inspections carried out for three different Danish water utilities companies. This dataset is used for training and testing decision tree methods and convolutional neural networks (CNNs) for automatic water level estimation. We pose the estimation problem as a classification and regression problem, and compare the results of both approaches. Furthermore, we compare the effect of using different inspection standards for labeling the ground truth water level. By treating the problem as a classification task and using the 2015 Danish sewer inspection standard, where water levels are clustered based on visual appearance, we achieve an averaged F1 score of 79.29% using a fine-tuned ResNet-50 CNN. This shows the potential of using CNNs for water level estimation. We believe including temporal and contextual information will improve the results further.

1 Introduction

Sewer networks are a critical piece of infrastructure that allow safe transportation of wastewater from households to specialized treatment plants. Sewer pipes are built for transporting either rain water, waste water, or a combination of both. In Germany, there are nearly 600,000 kilometers of public sewer pipes [1]. In the US, it has been estimated that the length of public net of sewers extends to over 1.2 million kilometers [2]. Because the sewer pipes are buried beneath roads and streets, their presence is easy to forget—until they break down. Replacement of an entire sewer pipe is costly and can require a large excavation work that entails disruptions to the road traffic. A more economical option is to refurbish the pipes before they break down [3], but this requires knowledge of the condition of the pipes. However, sewer pipes are difficult to inspect as most pipes are not accessible by human inspectors due to their small diameters. For large diameter pipes, the presence of toxic gases and the general contents of the sewage water renders the inspection a safety and health risk to human workers. The most common method for estimating the condition of a sewer pipe is to use tethered robots that are inserted into the pipe from the nearest accessible well. The inspection robot is typically equipped with a Closed-Circuit Television camera (CCTV) and a light source. A human operator controls the robot from a specially designed van and manually assesses the incoming video data. The overall inspection procedure is slow and prone to human errors. Therefore, there is a large research and industrial interest



(a) Tilted viewpoint. (b) Reflections. (c) Mist. (d) Infiltration.

Fig. C.1: Examples of adverse conditions from Closed-Circuit Television camera (CCTV) footage of sewer pipes which can complicate the water level classification task; (**a**) the camera is tilted such that the entire view is placed above the water line; (**b**) the water surface reflects the surroundings; (**c**) gases in the sewer impair the visibility in the same way as mist and haze; (**d**) infiltrating water.

in automating the inspection process through the use of computer vision and machine learning [4].

Accessibility of the sewer for inspection by a robotic platform is one of the most fundamental problems the inspection has to address. The accessibility is linked to the amount of water present in the pipe. In order to detect and classify defects in the pipe, substantial portions of the pipe must be visible above the water level. In short, the water level is a key indicator of how much of a pipe can be inspected. According to the European sewer inspection standards (EN 13508-2) [5], estimating the water level of the sewer pipe is therefore of paramount importance to human inspectors. Sample footage from inspection data is shown in Figure C.1. At first sight, estimating the water level from sewer pipes might appear to be a straightforward task for a computer vision algorithm to solve. The sewer is a confined space with few interruptions from the outside. However, the nature of the pipes and their contents renders a range of problems. The only source of light comes from the inspection vehicle, and some portions of the pipe might therefore not be sufficiently lit. Toxic gases are commonplace and renders as mist or fumes, resulting in a hazy image with reduced information. The presence of water entails different levels of reflection and might even flood the entire view. As a result, dust and sewage might stick to the lens and severely impair the visibility.

While there has been a lot of prior work on sewer defect classification and detection, as surveyed by Haurum and Moeslund [4], few researchers have worked specifically with water level estimation. Prior work utilizes classical computer vision techniques and modern deep learning-based methods. Classic computer vision methods have been used to detect key features of the sewer pipe which can be used to detect the water line [6, 7]. Recently, Convolutional Neural Network (CNN)-based methods have gained interest in automated defect classification [7–9], wherein some researchers have included water stagnation [8] and “high water levels” [9]. Most importantly, a recent paper [10] proposed training a deep learning-based segmentation model which segments the water in the image and infers the water lines. The researchers collected 1440 high-resolution RGB images, of which 167 were used for human evaluation and 80 images were used for training. The authors claim to achieve a perfect segmentation of the water level, beating manual and traditional computer vision annotation methods, and being able to calculate the flow rates and velocities by applying Manning's equation. However, their dataset only consists of recordings from a single pipe acquired over a 24-h period. This does not nearly capture the amount of variability encountered when inspecting real sewers. Furthermore, the high image resolution is not representative for sewer inspection CCTV videos which rarely exceeds full HD.

Several vision-based approaches for water level estimation have been proposed within the wastewater flow estimation field. A common technique is to measure the depth from a stationary staff gauge [11–15]. Nguyen et al. [11] and Jeanbourqin et al. [12] proposed to use an infrared camera to locate the water–air intersection line on the staff gauge using computer vision techniques. Handheld devices [15] and calibrated cameras [13, 14] have also been used for automated staff gauge readings. Alternatively, Sirazitdinova et al. [16] determined the water level using a stereo camera setup, while Khorchani and Blanpain [17] used a single calibrated camera. A common characteristic among these methods is the need for a stationary object, being either the camera setup or the staff gauge.

On the contrary, we investigate the feasibility of estimating the water levels in realistic and unseen sewer inspection videos by the use of a single input image at a time, from a moving uncalibrated camera. We compare both decision tree methods and deep learning-based methods in order to determine whether the extra complexity introduced with the neural networks is justified. Furthermore, as shown from the examples of Figure C.1, the inspection imagery is distinct from commonplace computer vision datasets such as ImageNet [18] or COCO [19]. Therefore, we investigate the effect of ImageNet pre-training compared to training from scratch on the available data. Our contributions are the following.

- We show that it is possible to reliably estimate the water level in unseen sewer pipes using a classification-based CNN.
- We show the evaluation performance impact of how the water levels are categorized, using the Danish sewer inspection standards as a use case.
- We show that CNN-based methods outperform traditional decision tree-based methods for water level estimation.
- We open source our model and analysis code (`https://bitbucket.org/aauvap/waterlevelestimation`).

The remainder of the article is structured as follows. In Section 2, we introduce our dataset. In Section 3, we describe the proposed methods, loss functions, and the training procedure. In Section 4, we detail the evaluation metrics and present the experimental results. In Section 5, we analyze and discuss the presented results. Finally, in Section 6 we summarize our findings and possible future directions.

2 Dataset

As made clear in [4], there are currently no publicly available sewer datasets. Therefore, we utilize our own dataset consisting of CCTV recordings from actual sewer inspections conducted for three different water utility companies across Denmark.

2.1 Dataset Construction

Professional sewer inspectors have assessed the data in real-time and provided manual annotations of the water level based on expert assessments. The data have been annotated following the 2010 Danish sewer inspection standards [20], where the water level is annotated in discrete steps of 10 in the interval [0, 100]. For example, if a sewer is annotated as having a water level of 40%, then the actual water level is somewhere between 35% and 45%, as illustrated in Figure C.2.

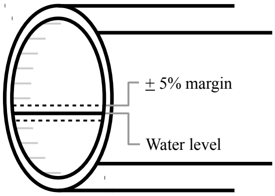


Fig. C.2: Illustration of the 5% point uncertainty margin allowed in the annotations. The gray horizontal lines in the left side of the pipe indicate the discrete steps from 0% to 100%. A water level of 40% is present in the given example.

The dataset used in this research is constructed from 5511 different CCTV videos where one or more images are extracted from each video, resulting in a total of 11,558 images. The large amount of videos is necessary to represent the variability of real-world sewer pipes. Even for skilled sewer inspectors it can be extremely difficult to estimate the water level. From our study of the recorded data, and from conversations with the water utility companies, we have found that there are often variations in the water level within the recordings. Therefore, noise is expected in the annotations.

We split the dataset into three parts: training, validation, and test. The dataset has been carefully constructed such that each annotation level is equally represented. Therefore, we sample 1000 images per class for the training split and 100 images for each of the validation and testing splits. However, there are not 1200 occurrences across the available data for the water levels between 70% and 100%. In these cases, we use the available data and note that the dataset is imbalanced for those classes. The distribution of images with respect to the splits can be seen in Table C.1. The data in the training, validation, and test splits all come from unique sewer pipes and inspections.

While the data are originally annotated using the 2010 Danish sewer inspection standards, the standards have been updated in 2015 [21]. In the newer inspection standards, the annotation protocol for water levels has been changed such that it focuses

Table C.1: Overview of the dataset and the three splits. The data are annotated following the 2010 Danish sewer inspection standards.

Water Level	Training		Validation		Test		Total	
	Images	Videos	Images	Videos	Images	Videos	Images	Videos
0%	1000	966	100	98	100	96	1200	1160
10%	1000	972	100	97	100	98	1200	1167
20%	1000	841	100	88	100	90	1200	1019
30%	1000	696	100	66	100	83	1200	845
40%	1000	574	100	55	100	59	1200	688
50%	1000	494	100	58	100	50	1200	602
60%	1000	361	100	33	100	43	1200	437
70%	531	181	97	31	31	17	659	229
80%	718	211	85	28	64	28	867	267
90%	257	79	80	12	100	11	437	102
100%	1000	199	100	22	95	23	1195	244
Total	9506	4529	1062	492	990	490	11,558	5511

on a coarse grouping of water levels instead of fine-grained intervals. Specifically, the water level is annotated into four classes: less than 5%, between 5% and 15%, between 15% and 30%, and above 30%. These groupings better correspond to the visual appearance of the water level, as it can be hard to distinguish the classes of more than 30% due to the inspection camera being partially or fully submerged under water. In these cases, the inspectors have previously used contextual and temporal information in order to complete their annotations.

It is possible to make a near perfect one-to-one mapping between the two standards with the exception being the 2015 class with $\geq$30% water level. As the 2010 annotations have a ±5% point uncertainty margin, the 30% class may contain data from water levels as low as 26%. We choose to accept the risk of this and acknowledge it as a source of extra noise. With this mapping, the dataset presented in Table C.1, annotated following the 2010 Danish sewer inspection standards, can be re-labeled into a dataset following the 2015 Danish sewer inspection standards as shown in Table C.2. Furthermore, by converting the data to the 2015 standards, the dataset becomes much more skewed towards the $\geq$30% class. This causes the dataset to be heavily imbalanced, compared to the dataset following the 2010 standards. However, we do not resample the data to be balanced for the 2015 standards as it will cause the experiments to be incomparable due to different training sets.

2.2 Data Quality

As the data are from real-life sewer inspections, the video resolution and quality vary depending on the utilized recording equipment. All the data have, however, been recorded with 25 frames per second. Furthermore, all of the videos have a text layer applied with inspection metadata, annotations, and more. This information has been blurred in order to ensure that the CNN-based methods learn by observing the

Table C.2: Overview of the dataset and the three splits. The data is annotated following the 2015 Danish sewer inspection standard.

Water Level Intervals	Training		Validation		Test		Total	
	Images	Videos	Images	Videos	Images	Videos	Images	Videos
WL < 5%	1000	966	100	98	100	96	1200	1160
5% ≤ WL < 15%	1000	972	100	97	100	98	1200	1167
15% ≤ WL < 30%	1000	841	100	88	100	90	1200	1019
30% ≤ WL	6506	1750	762	209	690	206	7958	2165
Total	9506	4529	1062	492	990	490	11,558	5511

water in the sewer pipe and not by reading the textual metadata. A range of examples from the dataset are shown in Figure C.3.

3 Methodology

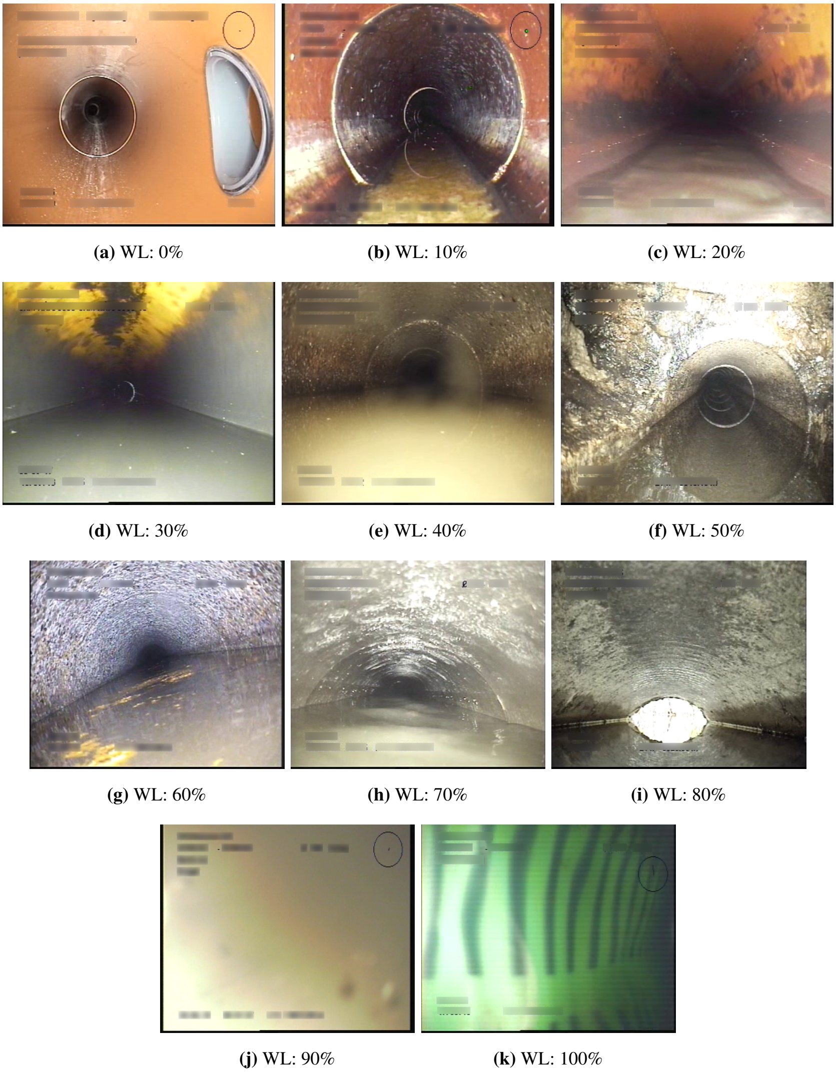
We investigate the performance of multiple machine learning methods using both the 2010 and 2015 Danish annotation guidelines. First, we train the proposed models using the annotations following the 2010 standard in a classification approach where each level is a discrete class. Second, we train the models in a regression setting where the water level percentage is predicted as a continuous quantity. Last, we train the models in a classification setting and convert the annotations to the 2015 standard classes where the different classes are grouped as mentioned in Section 2. These settings are referred to as Class10, Reg2Class10, and Class15, respectively.

3.1 Features and Models

We investigate the performance of two CNNs—AlexNet and ResNet—to determine whether deep learning is feasible for estimating the water level in sewers from single images. Furthermore, by measuring the performance of ResNet-18, ResNet-34, and ResNet-50, we can evaluate the effect of higher abstraction levels provided by an increased network depth.

AlexNet [22] is considered to be the deep neural network that sparked the interest for deep learning almost ten years ago, and it is often used as a baseline for classification tasks. A neural network is considered deep when there is more than a single layer between the input and output layers, and AlexNet has eight such layers. Generally speaking, the feature abstraction level increases with the depth of the network which, in theory, means that a deeper network can handle more complex tasks. However, as the amount of parameters increases, so does the processing time and the likelihood of overfitting the model to the training data. It has even been shown that for some architectures it can harm the performance if the depth is overly increased due to the degradation problem [23].

ResNet [24] is a family of CNN models developed with the aim of being able



(a) WL: 0%

(b) WL: 10%

(c) WL: 20%

(d) WL: 30%

(e) WL: 40%

(f) WL: 50%

(g) WL: 60%

(h) WL: 70%

(i) WL: 80%

(j) WL: 90%

(k) WL: 100%

Fig. C.3: Images from the dataset that show the inter-class variation. WL is the annotated water level.

to utilize the increased abstraction level offered by deeper layers without suffering from the degradation problem. The models consist of stacks of relatively small layers connected by identity shortcuts that forces the network to learn the residual function between the stacks. These shortcuts allow the networks to cheaply reduce the influence of certain layers if they do not enhance the output performance. This type of architecture has proven to be very powerful and ResNets are still widely used; especially for cases where depth is expected to improve the performance.

Two decision tree methods—Random Forest [25] and Extra Trees [26]—are also investigated in order to provide a baseline performance. The tree-based methods are trained using GIST features [27] computed from the images as Myrans et al. The authors of [28] have shown this to be an effective combination for sewer defect classification. The GIST feature descriptor [27] applies a series of 2D Gabor filters, each with a different scale and orientation, resulting in a feature map per scale and orientation permutation. The feature maps are divided into a predefined grid where the feature values within each grid element are averaged. The averaged feature values are then concatenated per feature map into a feature vector, and all of the resulting feature vectors are concatenated to give the final GIST feature vector.

3.2 Loss Functions

The classification models are all evaluated using the standard categorical cross-entropy (CE) loss with the option of including class specific weights, w_c. The cross entropy loss is defined as

$$f_{CE} = -\sum_{c=1}^{C} w_c y_c \log(p_c) \tag{C.1}$$

where y_c is the ground truth label, 1 if the correct class, and 0 otherwise, and p_c is the predicted probability of class c. For the standard CE loss, w_c is set to 1 for all classes.

However, for the regression networks there is not a single standard loss. A large set of methods utilizes the Mean Absolute Error (MAE) or Mean Square Error (MSE) loss functions, also known as the ℓ_1 and ℓ_2 loss functions, respectively. MAE and MSE are defined as

$$f_{\text{MAE}}(x) = |x| \tag{C.2}$$

$$f_{\text{MSE}}(x) = x^2 \tag{C.3}$$

where x is the residual, the result after subtracting the predicted value from the ground truth value.

The MAE loss function is robust to outliers but suffers from derivatives that are not continuous. MSE is more stable during training due to continuous derivatives, but more sensitive to outliers due to the squared residual term. Due to the built-in $\pm 5\%$ uncertainty around each annotation in the 2010 standards, we choose to train with the MSE loss as the quadratic residual term allows automatically increasing the weighing the further away the prediction is from the ground truth.

3.3 Training Procedure

The CNNs are all trained using the hyperparameters stated in Table C.3. The Adam optimizer [29] is chosen as it continuously adapts the learning rate for each parameter based on the first- and second-order moments of the gradients. The initial learning rate is set to 0.01 for models learned from scratch whereas a reduced learning rate of 0.001 is used for fine-tuning networks pretrained on the ImageNet dataset. When fine-tuning the ResNet models, we freeze the first two residual blocks in order to retain low-level

Table C.3: Hyperparameters for each of the Convolutional Neural Network (CNN) models.

Parameter	**From Scratch**	**Fine-Tuned**
Learning Rate	0.01	0.001
Weight Decay	0.0001	
Optimizer	Adam	
Batch Size	64	
Epochs	50	

Table C.4: Hyperparameter search intervals for the Random Forest and Extra Trees algorithms. *d* denotes the dimensionality of the input features for GIST.

Parameter	**Values**
Number of Trees	[10, 100, 250]
Maximum Depth	[10, 20, 30]
Maximum Features	$[\sqrt{d}, \log_2(d), d/3, d]$

feature knowledge. A weight decay of 0.0001 is used for all models to help regularize the weight parameters and avoid overfitting. All models are trained for 50 epochs with a batch size of 64 to make them comparable. During training, the input images are augmented by horizontally flipping the image with a 50% chance. All images during both training and evaluation are normalized.

We conduct a small hyperparameter search in order to find the best set of parameters for the tree-based methods. The investigated hyperparameters and the possible values are shown in Table C.4, where *d* is the amount of features in the GIST feature descriptor. For the classification models, the minimum number of samples required to be at a leaf node is set to 1, whereas for the regression models it is set to 5, as per the original Random Forest paper [25]. The GIST feature descriptor is computed using a 4×4 grid with filters using 4 scales and 8 orientations. The input image is downscaled to 128×128 pixels and converted to grayscale as described in [28], which results in a 512 dimensional GIST feature vector.

All classification models are trained by minimizing a weighted CE loss where the class weights are calculated as

$$w_c = \frac{\max(N_i)}{N_c}, \; i \,\forall\, [1,2,...,C], \tag{C.4}$$

where N_i is the number of training samples for class i and w_c is the weight for class c, out of the total C classes.

The regression models are trained by minimizing the MSE loss. The best performing model is determined by selecting the model with the lowest validation loss. For the CNNs, the validation loss is computed after each epoch. The best performing

Table C.5: Hyperparameters for each of the best performing Random Forest and Extra Trees models on each task.

Parameter\Model	Random Forest			Extra Trees		
	Class10	**Reg2Class10**	**Class15**	**Class10**	**Reg2Class10**	**Class15**
Number of Trees	250	250	250	250	250	250
Maximum Depth	10	20	10	10	20	10
Maximum Features	$\sqrt{d}$	d	$\sqrt{d}$	d	d	d

tree-based models are found from the model with the lowest validation loss among the models in the aforementioned hyperparameter search, see Table C.5.

All of the CNN architectures are implemented using the PyTorch framework [30], utilizing the publicly available implementations as well as the provided network weights for the ImageNet pretrained models. The models were trained on a single RTX 2080 TI graphics card. For the tree-based models, we utilize the Scikit-Learn library [31] while the GIST features are extracted using an open source Python wrapping of the original C implementation [32].

4 Experimental Results

We observe that in general the fine-tuned CNNs outperforms the CNNs trained from scratch, indicating that while the ImageNet dataset is visually quite far from the sewer image data the learned information is still valuable. This aligns with prior experience within the transfer learning domain where ImageNet pretraining is the norm. Therefore, we only show the performance of the fine-tuned CNNs in Tables C.6–C.9. The results of the CNNs trained from scratch can be found in Appendix C.A.

Performance Metrics

The tasks are evaluated using the F1-metric which is calculated as the harmonic mean of the Precision, P, and the Recall, R, of the predictions, as shown in Equations (C.5)–(C.7). TP, FP, and FN denote the true positive, false positive, and false negative predictions, respectively.

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{C.5}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{C.6}$$

$$\mathrm{F1} = \frac{2\,\mathrm{P}\,\mathrm{R}}{\mathrm{P} + \mathrm{R}} \tag{C.7}$$

As the task at hand is a multi-class classification problem, we generalize the binary F1-metric by calculating the average of the F1-metric for each class. This is done by calculating the micro- and macro-averaged F1-metrics. These F1-metrics are chosen as the normal accuracy metric is not representative for imbalanced data. Furthermore, the two F1-metrics incorporate an implicit weighting for minority and majority classes such that different trends for the imbalanced dataset are highlighted.

The micro-F1 metric is calculated by treating all observations equally, resulting in a metric sensitive to the majority classes. Micro-F1 is calculated based on the micro-averaged precision and recall, as shown in Equations (C.8)–(C.10), where C denotes the amount of classes. TP_c, FP_c, and FN_c are the binary metrics for class c, obtained by approaching the evaluation as a one vs. all binary task for class c. Specifically, this means that the precision, recall, and F1-metric are calculated by globally counting all true positive, false positive, and false negative predictions.

$$\text{micro-P} = \frac{\sum_{c=1}^{C} \mathrm{TP}_c}{\sum_{c=1}^{C} \mathrm{TP}_c + \mathrm{FP}_c} \tag{C.8}$$

$$\text{micro-R} = \frac{\sum_{c=1}^{C} \mathrm{TP}_c}{\sum_{c=1}^{C} \mathrm{TP}_c + \mathrm{FN}_c} \tag{C.9}$$

$$\text{micro-F1} = \frac{2 \text{ micro-P micro-R}}{\text{micro-P} + \text{micro-R}} \tag{C.10}$$

The Macro-F1 metric is calculated as the arithmetic mean of the per-class F1-metrics as shown in Equation (C.11). This results in an equal weighting for each class, thereby causing the Macro-F1 metric to be more sensitive to the rare classes.

$$\text{Macro-F1} = \frac{1}{C} \sum_{c=1}^{C} \mathrm{F1}_c \tag{C.11}$$

In order to compare the regression models with the classification models, we convert the regression output to classification outputs for training setting Reg2Class10. This is achieved by utilizing the fact that each class represents a ±5% point interval around the center value. The regression predictions are therefore converted by first clamping the values to the interval $[0, 100]$ and subsequently assigning the regression prediction to the closest ground truth value. For instance, a regression output of 74.5% water level will be assigned to the 70% level class. The regression outputs are not converted to the Class15 labels, as the conversion cannot be performed without uncertainty near the 30% water level area. The results for all methods are shown in Table C.6 and the F1 score for each class under the different training variations are shown in Tables C.7–C.9 and Tables C.10–C.13. Models trained from scratch are indicated with a “S” suffix, while fine-tuned models have a “FT” suffix. The per-class performance is also visualized in Figure C.4.

Table C.6: Results for each tested method for the different training settings.

Method	**Class10**		**Reg2Class10**		**Class15**	
	micro-F1	**Macro-F1**	**micro-F1**	**Macro-F1**	**micro-F1**	**Macro-F1**
Random Forest	27.17	23.19	14.63	11.01	68.18	51.47
Extra Trees	29.49	26.39	14.33	10.72	64.34	50.19
AlexNet-FT	30.10	26.96	30.10	28.81	69.59	20.54
ResNet18-FT	39.19	**37.41**	**30.61**	**30.00**	73.03	60.93
ResNet34-FT	37.37	35.54	28.69	28.00	76.36	61.88
ResNet50-FT	**39.70**	36.50	27.07	26.27	**79.29**	**62.88**

Table C.7: Per-class F1 score for each method—Class10 training setting.

Method\Water Level	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Random Forest	48.25	32.58	29.38	19.32	22.11	21.52	17.89	3.45	16.13	9.43	35.02
Extra Trees	45.05	36.65	33.79	19.88	27.84	26.09	20.32	6.35	13.14	28.33	32.85
AlexNet-FT	49.38	41.75	28.05	31.54	15.60	25.32	14.77	10.77	17.27	49.61	12.50
ResNet18-FT	**71.72**	55.67	32.05	16.67	**34.10**	**31.43**	24.62	10.89	**26.23**	**66.94**	41.18
ResNet34-FT	68.38	**57.65**	**39.52**	**34.38**	26.29	15.04	23.79	11.90	21.36	59.72	32.91
ResNet50-FT	63.32	47.49	38.20	23.96	31.30	13.11	**29.63**	**13.86**	14.74	66.08	**59.77**

Table C.8: Per-class F1 score for each method—Reg2Class10 training setting.

Method\Water Level	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Random Forest	0.00	20.55	14.71	20.14	26.51	19.53	13.20	1.41	5.13	0.00	0.00
Extra Trees	0.00	15.83	13.59	20.00	25.57	20.97	12.24	0.00	9.71	0.00	0.00
AlexNet-FT	53.89	41.79	33.65	**30.05**	25.11	24.56	23.81	9.02	16.51	**56.79**	1.69
ResNet18-FT	50.35	**49.04**	34.00	26.54	26.36	25.31	26.73	11.94	16.55	49.06	**14.16**
ResNet34-FT	**54.09**	40.84	**34.55**	23.66	23.70	**30.08**	26.84	11.68	**23.13**	34.90	4.55
ResNet50-FT	44.59	35.68	28.14	23.65	**26.96**	17.56	**27.59**	**12.58**	15.91	56.32	0.00

Table C.9: Per-class F1 score for each method—Class15 training setting.

Method\Water Level	**WL $<$ 5%**	**5% $\leq$ WL $<$ 15%**	**15% $\leq$ WL $<$ 30%**	**30% $\leq$ WL**
Random Forest	50.61	40.68	32.08	82.52
Extra Trees	47.15	39.18	35.16	79.28
AlexNet-FT	0.00	0.00	0.00	82.14
ResNet18-FT	**70.75**	48.62	38.41	85.94
ResNet34-FT	69.06	43.59	**46.36**	88.53
ResNet50-FT	65.38	**53.39**	41.24	**91.51**

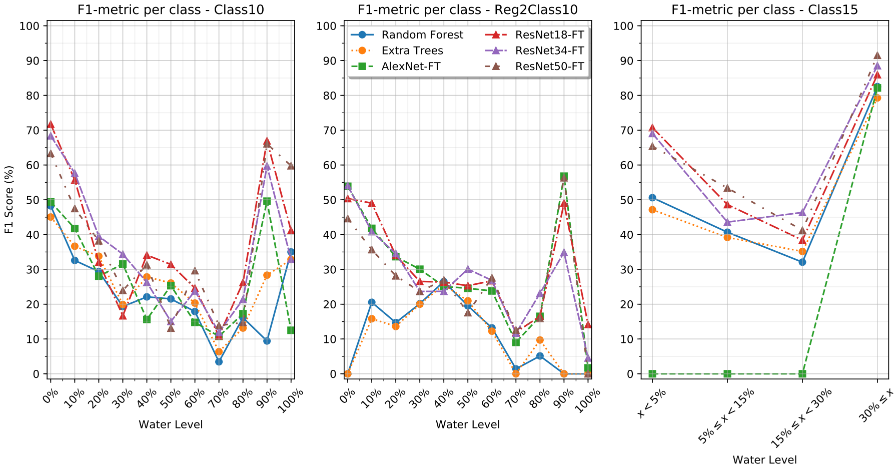


Fig. C.4: Per-class F1-metric for all methods under the different training settings.

5 Discussion

From the results presented in Table C.6 it is possible to see several trends. We observe that utilizing the 2015 classification scheme leads to a direct increase in performance compared to the 2010 standard. Specifically, we see that for all models, except AlexNet-FT, the F1-metrics have been improved dramatically. This corresponds with prior research by Van der Steen et al. [33], who found that the more detailed a sewer inspection standard is, the more mistakes inspectors make.

When looking into the training settings, we see that the classification approaches consequently match or outperform the methods trained with a regression approach. This indicates that the strict discrete class membership enforced by the classification approaches leads to better generalization than the soft continuous class membership enforced by the regression approaches. This may be a direct consequence of all the ground truth labels being discrete values with a known uncertainty margin, leading to the values actually representing a span of values. The regression approaches may, however, perform better if the ground truth labels were provided by a continuous measurement such as data from a flow meter.

When looking into the class-specific performance of the Class10 and Reg2Class10 training settings in Tables C.8 and C.9, we observe that the models have a high F1-metric for the first few classes where the water level is still visually distinguishable. However, as the water level increases, the F1 metric performance decreases until an increase in performance for the 90% and 100% water level classes. We see that for the tree-based methods in the Reg2Class10 setting, there are several classes with an F1-metric of 0% while other classes have an F1 metric of up to 26%. Similarly, we observe that the AlexNet-S model simply focuses on a single class in its predictions as also shown by the Macro-F1 score, while the AlexNet-FT model is capable of producing more meaningful predictions. We also see that the depth of the networks does not seem to correspond with an increase in performance of the F1-metric.

It is observed that the tree-based methods do not match the micro-F1 score of the ResNets and AlexNet for the Class15 training setting. However, when comparing the Macro-F1 metric, it is obvious that the tree-based methods outperform the AlexNet on some classes. By looking into the results in Table C.9 we see that the tree-based models perform well on the two extreme classes, $<5\%$ and $\geq 30\%$, but are not as capable at classifying the two intermediate classes where the inter-class variance may be more subtle. Moreover, we see that the AlexNets do not generalize at all, instead simply collapsing to predict only the majority class. This is in contrast with how the AlexNets performed in the Class10 and Reg2Class10 training settings, where only AlexNet-S failed to produce meaningful predictions.

These results show that by framing the water level estimation task as a clustering of perceptible amount of water, as in the 2015 Danish standards, better facilitates machine learning-based methods than using a direct mapping such as in the 2010 Danish standards. However, the results are not perfect, as there are still some classes with a low classification rate. This could potential be improved by including temporal information in the models, such that transitions between water levels can be detected and spurious classification be ignored. Such an approach has been applied by the authors of [28], who applied a Hidden Markov Model and window filtering to sewer defect classifications. Geometric information, such as the size and shape of the pipe, may also prove useful as these are closely linked with the water level. Last, information about defects would also help guide the models toward the correct water level classification. Defects such as pipe collapse or large roots could lead to abnormally high water levels.

6 Conclusions

Estimating the water level in sewers during inspection is important as it indicates the portion of the pipe that cannot be visually inspected. Currently, it is a subjective and difficult task of the inspector to estimate the water level through CCTV recordings and only limited research has been conducted on automating this process. A professionally annotated dataset with 11,558 CCTV sewer images provided by three Danish utility companies is used as the foundation for an investigation on the feasibility of using deep neural networks for automating water level estimation. The problem is studied through two classification tasks following the 2010 and 2015 Danish Sewer Inspection Standards. Four deep neural network models (AlexNet, ResNet-18, ResNet-34, and ResNet-50) and two traditional decision tree methods (Random Forest and Extra Trees) are compared against each other.

The deep learning methods generally outperform the decision trees, but the networks do not seem to benefit from the abstraction levels of the very deep layers. The best results are provided by ResNet with micro-F1 scores of 39.70% and 79.29%

following the 2010 and 2015 standards, respectively. These are promising results given that the data are noisy and the classifications are based on single images. Utilizing temporal, contextual, and geometric information could improve the classification rate and should be considered for future work.

Author Contributions: Conceptualization, J.B.H.; methodology, J.B.H. and M.P.; software, J.B.H.; validation, J.B.H., C.H.B., M.P., and T.B.M.; formal analysis, J.B.H.; investigation, J.B.H.; resources, J.B.H. and C.H.B.; data curation, J.B.H. and M.P.; writing—original draft preparation, J.B.H, C.H.B., and M.P.; writing—review and editing, J.B.H., C.H.B., M.P., and T.B.M.; visualization, J.B.H., C.B.H., and M.P.; supervision, T.B.M.; project administration, J.B.H. and T.B.M.; funding acquisition, T.B.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innovation Fund Denmark (grant number 8055-00015A) and is part of the Automated Sewer Inspection Robot (ASIR) project.

Conflict of Interest: The authors declare no conflict of interest.

C.A Results for CNNs Trained From Scratch

Table C.10: Results for the CNNs trained from scratch, for each of the different training settings.

Method	**Class10**		**Reg2Class10**		**Class15**	
	micro-F1	**Macro-F1**	**micro-F1**	**Macro-F1**	**micro-F1**	**Macro-F1**
AlexNet-S	10.05	1.67	10.05	1.67	69.59	20.54
ResNet18-S	25.96	23.59	18.79	17.79	70.81	**54.41**
ResNet34-S	**29.90**	25.72	**20.71**	**19.80**	**72.02**	53.35
ResNet50-S	29.29	**26.20**	19.79	18.92	68.18	48.31

Table C.11: Per-class F1 score for the CNNs trained from scratch—Class10 training setting.

Method\Water Level	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
AlexNet-S	0.00	0.00	18.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ResNet18-S	**55.45**	**41.67**	18.18	8.20	**27.07**	20.83	8.00	**16.04**	10.53	16.20	37.31
ResNet34-S	51.68	38.64	**30.59**	**20.00**	16.54	**24.18**	11.76	11.49	8.85	22.62	**46.62**
ResNet50-S	43.22	32.00	28.24	10.37	24.51	14.93	**17.89**	6.11	**13.89**	**56.00**	40.99

Table C.12: Per-class F1 score for the CNNs trained from scratch—Reg2Class10 training setting.

Method\Water Level	**0%**	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
AlexNet-S	0.00	0.00	0.00	0.00	0.00	18.35	0.00	0.00	0.00	0.00	0.00
ResNet18-S	**33.33**	**37.56**	18.10	20.87	**17.92**	**23.04**	16.43	11.01	11.37	3.97	**2.04**
ResNet34-S	20.69	36.84	**28.18**	20.72	12.67	21.05	13.98	**16.42**	**12.97**	32.22	**2.04**
ResNet50-S	18.64	26.83	19.81	**23.85**	16.55	22.86	**17.02**	7.75	2.68	**52.11**	0.00

Table C.13: Per-class F1 score for the CNNs trained from scratch—Class15 training setting.

Method\Water Level	WL $<$ 5%	5% $\leq$ WL $<$ 15%	15% $\leq$ WL $<$ 30%	30% $\leq$ WL
AlexNet-S	0.00	0.00	0.00	82.14
ResNet18-S	**53.08**	**45.57**	33.49	85.49
ResNet34-S	52.78	38.92	**34.78**	**86.92**
ResNet50-S	50.49	25.00	33.90	83.87

References

[1] Statistisches Bundesamt, "Öffentliche wasserversorgung und öffentliche abwasserentsorgung - strukturdaten zur wasserwirtschaft 2016," Statistisches Bundesamt, Tech. Rep., 2018, fachserie 19 Reihe 2.1.3.

[2] American Society of Civil Engineers (ASCE), "2017 infrastructure report card - wastewater," 2017, accessed 2020-03-27.

[3] United States Environmental Protection Agency (EPA), "Fact sheet: Asset management for sewer collection systems," 2002.

[4] J. B. Haurum and T. B. Moeslund, "A survey on image-based automation of cctv and sset sewer inspections," *Automation in Construction*, vol. 111, p. 103061, 2020.

[5] European Committee for Standardization (CEN), *EN 13508- 2: Conditions of drain and sewer systems outside buildings – Part 2: Visual inspection coding system.* European Committee for Standardization (CEN), 2003.

[6] S. Kirstein, K. Müller, M. Walecki-Mingers, and T. M. Deserno, "Robust adaptive flow line detection in sewer pipes," *Automation in Construction*, vol. 21, pp. 24 – 31, 2012.

[7] M. R. Halfawy and J. Hengmeechai, "Integrated vision-based system for automated defect detection in sewer closed circuit television inspection videos," *Journal of Computing in Civil Engineering*, vol. 29, no. 1, 2015.

[8] D. Li, A. Cong, and S. Guo, "Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification," *Automation in Construction*, vol. 101, pp. 199 – 208, 2019.

[9] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, "Automatic detection and classification of sewer defects via hierarchical deep learning," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1836–1847, 2019.

[10] H. Ji, S. Yoo, B.-J. Lee, D. Koo, and J.-H. Kang, "Measurement of wastewater discharge in sewer pipes using image analysis," *Water*, vol. 12, no. 6, p. 1771, Jun 2020.

[11] L. S. Nguyen, B. Schaeli, D. Sage, S. Kayal, D. Jeanbourquin, D. A. Barry, and L. Rossi, “Vision-based system for the control and measurement of wastewater flow rate in sewer systems,” *Water Science and Technology*, vol. 60, no. 9, pp. 2281–2289, 11 2009.

[12] D. Jeanbourquin, D. Sage, L. Nguyen, B. Schaeli, S. Kayal, D. A. Barry, and L. Rossi, “Flow measurements in sewers based on image analysis: automatic flow velocity algorithm,” *Water Science and Technology*, vol. 64, no. 5, pp. 1108–1114, 09 2011.

[13] Y.-T. Lin, Y.-C. Lin, and J.-Y. Han, “Automatic water-level detection using single-camera images with varied poses,” *Measurement*, vol. 127, pp. 167 – 174, 2018.

[14] T. E. Gilmore, F. Birgand, and K. W. Chapman, “Source and magnitude of error in an inexpensive image-based water level measurement system,” *Journal of Hydrology*, vol. 496, pp. 178 – 186, 2013.

[15] M. Bruinink, A. Chandarr, M. Rudinac, P. van Overloop, and P. Jonker, “Portable, automatic water level estimation using mobile phone cameras,” in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*, 2015, pp. 426–429.

[16] E. Sirazitdinova, I. Pesic, P. Schwehn, H. Song, M. Satzger, M. Sattler, D. Weingärtner, and T. M. Deserno, “Sewer discharge estimation by stereoscopic imaging and synchronized frame processing,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 7, pp. 602–613, 2018.

[17] M. Khorchani and O. Blanpain, “Free surface measurement of flow over side weirs using the video monitoring concept,” *Flow Measurement and Instrumentation*, vol. 15, no. 2, pp. 111 – 117, 2004.

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 4 2015.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “t,” in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.

[20] Dansk Vand og Spildevandsforening (DANVA), *Fotomanualen: TV-inspektion af afløbsledninger*, 6th ed. Dansk Vand og Spildevandsforening (DANVA), 2010.

[21] ——, *Fotomanualen: TV-inspektion af afløbsledninger*, 7th ed. Dansk Vand og Spildevandsforening (DANVA), 2015.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.

[23] K. He and J. Sun, “Convolutional neural networks at constrained time cost,” 2014.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[25] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[26] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 3 2006.

[27] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[28] J. Myrans, R. Everson, and Z. Kapelan, “Automated detection of faults in sewers using cctv image sequences,” *Automation in Construction*, vol. 95, pp. 64 – 71, 2018.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d´ Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[31] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[32] Y. Tsuchiya, “lear-gist-python,” 2020. [Online]. Available: https://github.com/tuttieee/lear-gist-python

[33] A. J. van der Steen, J. Dirksen, and F. H. Clemens, “Visual sewer inspection: detail of coding system versus data quality?” *Structure and Infrastructure Engineering*, vol. 10, no. 11, pp. 1385–1393, 2014.

Paper D

Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder

Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, and Thomas B. Moeslund

The paper has been published in the
Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) pp. 1441–1452, 2022.

The layout has been revised.

Abstract

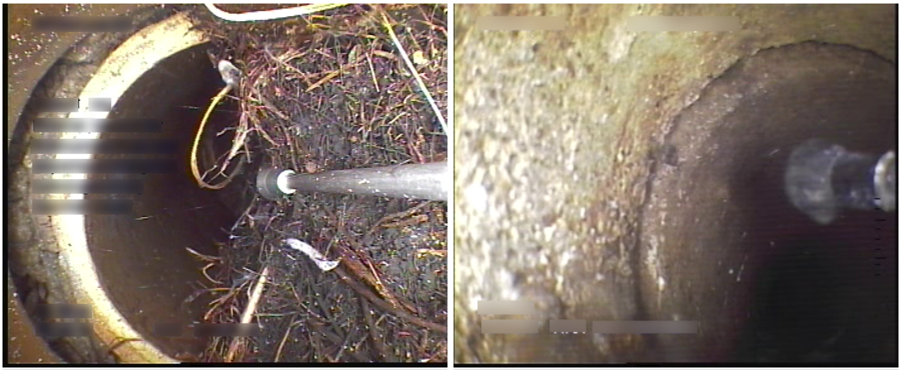
The sewerage infrastructure is one of the most important and expensive infrastructures in modern society. In order to efficiently manage the sewerage infrastructure, automated sewer inspection has to be utilized. However, while sewer defect classification has been investigated for decades, little attention has been given to classifying sewer pipe properties such as water level, pipe material, and pipe shape, which are needed to evaluate the level of sewer pipe deterioration.

In this work we classify sewer pipe defects and properties concurrently and present a novel decoder-focused multi-task classification architecture Cross-Task Graph Neural Network (CT-GNN), which refines the disjointed per-task predictions using cross-task information. The CT-GNN architecture extends the traditional disjointed task-heads decoder, by utilizing a cross-task graph and unique class node embeddings. The cross-task graph can either be determined a priori based on the conditional probability between the task classes or determined dynamically using self-attention. CT-GNN can be added to any backbone and trained end-to-end at a small increase in the parameter count. We achieve state-of-the-art performance on all four classification tasks in the Sewer-ML dataset, improving defect classification and water level classification by 5.3 and 8.0 percentage points, respectively. We also outperform the single task methods as well as other multi-task classification approaches while introducing 50 times fewer parameters than previous model-focused approaches. The code and models are available at the project page `http://vap.aau.dk/ctgnn`*.*

1 Introduction

The sewerage infrastructure is a key infrastructure of modern society, which needs to be regularly inspected and maintained in order to ensure its functionality [2]. These inspections require professional sewer inspectors who are capable of documenting and differentiating the fine-grained sewer defects, but also the properties of the sewer pipe such as the water level, pipe shape and pipe material, see Figure D.1. All of this information can be combined to compute a single deterioration score for each sewer pipe [3] used by water utility companies for asset management. Due to the hidden nature of the sewerage infrastructure sewer inspections are hard and cumbersome to conduct, as the sewer inspectors have to inspect using a remote controlled vehicle with a movable camera. Each inspection can stretch over a long duration of time due to obstacles in the sewers and limited speed of the vehicle. This leads to prolonged duration of looking at a screen, and can potentially result in flawed inspections due to fatigue.

In order to alleviate and assist the sewer inspectors, academia and industry have researched how to automate parts of the inspection process for more than 30 years [4]. However, the majority of work within this field has been focused on the important task of classifying the defects present in the pipes, while omitting the concurrent tasks of



Task	Ground Truth	R50-MTL	CT-GNN
Defect	FS, RO	FS	FS, RO
Water	[0%,5%)	[0%,5%)	[0%,5%)
Shape	Circular	Circular	Circular
Material	VC	VC	VC

Task	Ground Truth	R50-MTL	CT-GNN
Defect	OB, FS	FS	OB, FS
Water	[0%,5%)	[0%,5%)	[0%,5%)
Shape	Circular	Circular	Circular
Material	Conc.	VC	Conc.

Fig. D.1: Example images from the Sewer-ML dataset [1] together with examples showing how the baseline R50-MTL model with no cross-task relationship modeling misses the noticeable roots (RO) and surface damage (OB). Additionally, the R50-MTL model misclassifies the material as vitrified clay (VC) instead of as concrete (Conc.), whereas the proposed CT-GNN model classifies all classes in each task correctly in both examples.

determining the water level, pipe material, and pipe shape needed to determine the deterioration score [4]. Furthermore, as the inspections are performed on location it is infeasible to deploy several large models for each task.

Therefore, we investigate how to utilize Multi-Task Learning (MTL), and its subfield Multi-Task Classification (MTC), to simultaneously classify the sewer pipe defects and properties, by training a single model that is capable of processing multiple tasks during a single forward pass [5].

The MTC problem is often defined as learning how to solve several **unrelated** datasets with a single network [6, 7], whereas the problem of **related** and **concurrent** classification tasks, as *e.g.* during sewer inspections, is less well understood [1, 8]. The occurrence of the different task classes follows a hidden intractable joint distribution over all classes from all tasks. While the joint distribution is intractable, the co-occurrence information of the task classes can be inferred from the data, or learned by a model, and subsequently utilized to improve the classification process.

In order to handle the concurrent MTC problem, we propose a novel decoder-focused model, the Cross-Task Graph Neural Network (CT-GNN) Decoder, where the per-task features are refined using a cross-task sharing mechanism, inspired by recent dense vision decoder-focused models [9–12]. Specifically, we propose applying a CT-GNN on the initial task feature representations utilizing cross-task class relationships to refine the predictions.

We find that classification of all tasks can be improved by incorporating these cross-task class relationships into the decoder, by either utilizing the a priori known co-occurrence of the different task classes or dynamically estimating it through self-attention. Our proposed method is illustrated in Figure D.2. Compared to the previously limited use of graphs in MTC, we do not utilize feature vectors from different images in a batch [13, 14] nor do we consider sequential data inputs [15]. Compared to previous decoder-focused MTC models, we neither estimate the statistical relationship from batches [16], nor impose tensor-based constraints [17, 18].

Our contributions are therefore the following:

- We present the Cross-Task GNN Decoder, a novel MTC decoder that refines the per-task features through a late cross-task mechanism, trained in an end-to-end manner with only a small parameter count increase.
- In order to quantify a priori knowledge of task relationships we construct a cross-task graph adjacency matrix in a data-driven manner.
- We achieve State-of-the-Art performance on all four classification tasks in the Sewer-ML dataset [1], demonstrating the importance of utilizing cross-task relationships during automated sewer inspections.

The paper is structured as follows. In Section 2, we review the related works within the automated sewer inspection as well as MTL and MTC fields. In Section 3, we introduce the CT-GNN decoder head and how to construct the adjacency matrix. In Section 4, we compare the CT-GNN against other MTC methods on the Sewer-ML dataset, investigate per-class performances, and conduct ablation studies. Finally, in Section 5, we conclude the paper.

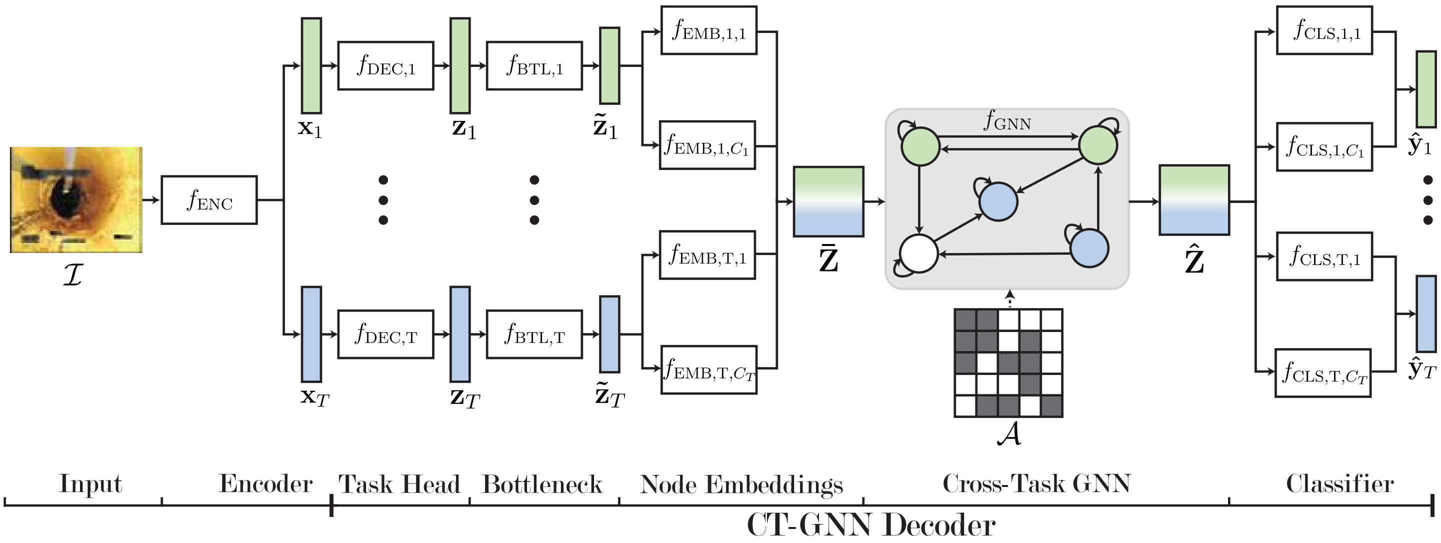


Fig. D.2: CT-GNN Overview. The proposed CT-GNN decoder and its location within the typical MTC architecture. The initial task features, $\mathbf{z}_t$, $t = 1, 2, \dots, T$, from T disjointed task-heads, are refined using our CT-GNN decoder, which incorporates class relationship knowledge, resulting in the final class predictions $\hat{\mathbf{y}}_t$, $t = 1, 2, \dots, T$. The CT-GNN is explained in detail in Sections 3.3 & 3.4.

2 Related Works

Automated Sewer Inspections. The field of automated sewer inspections has been researched for several decades by both academia and industrial research and development [4]. However, until the release of the Sewer-ML dataset [1] there was no public dataset or commonly agreed upon evaluation protocol [4].

The majority of work within the field has instead focused on automatically classifying defects using CCTV images [1] and other sensor based approaches [19–28]. Only within recent years [4] have deep learning based methods been utilized for defect classification [1, 29–34], detection [35–37], segmentation [38–41], and spatiotemporal based analysis [42, 43]. Defect classification models often employ a two-stage approach with a small initial classifier making a binary defect/non-defect classification, followed by a specialized defect classifier [1, 29, 31, 32, 34]. Recently, work has been conducted on classifying the water level in sewer pipes [44, 45], such that it is possible to estimate how much of the pipe can be inspected for defects. However, no work has been conducted on classifying the sewer pipe defects and properties concurrently. For an in-depth review of the vision-based automated sewer inspection field we refer to the survey by Haurum and Moeslund [4].

Multi-Task Learning. The field of multi-task learning has been applied across several different domains. Within the computer vision domain, MTL has been applied on image-level classification tasks such as facial attributes [8] and age and gender estimation [46, 47], learning several unrelated datasets at a time [6, 7], as well as learning multiple dense vision tasks such as per-pixel depth estimation and semantic segmentation [48–51]. Two main research branches have been developed through the years: optimization-focused and model-focused approaches [5]. For an exhaustive review of the field we refer to the surveys of the field [5, 52, 53].

The optimization-focused approaches investigate the effect of balancing how the tasks are learned. The tasks are balanced through operations such as normalizing the gradient magnitudes [54], approaching the problem as a multi-objective optimization problem and finding a Pareto optimal solution among all tasks [55, 56], adjusting the task weights based on the loss descent rate [57], the task-dependent homoscedastic uncertainties [58, 59], and more [60–62]. Each of these approaches is built on different underlying assumptions regarding how the task balancing is controlled, and introduces either an extra computational load or extra hyperparameters.

The model-focused approaches investigate the effect of parameter sharing in the model and is classically split into two types, hard and soft parameter sharing. Hard parameter sharing approaches are built around a shared backbone split into task-specific branches and heads [63–67], whereas in soft parameter sharing each task is assigned its own parameters with cross-task information introduced through one or more feature sharing mechanisms [57, 68–70]. Typically, these models utilize an encoder-decoder structure, where an input is passed through an encoder generating a global or per-task feature representation, which is used by a decoder to produce the task predictions [5].

This has led to encoder- and decoder-focused methods.

In encoder-focused models the task parameters are only shared in the encoder, while the decoder consists of disjointed task-heads with no cross-task information [54, 56, 58]. In decoder-focused models, the model parameters are also shared across tasks in the decoder through mechanisms such as multi-model distillation [9–12, 71], sequential task prediction [72], or cross-task consistency [73]. Decoder-focused models have been applied primarily for dense vision tasks. The few decoder-focused models that have been applied to multi-task classification depend on tensor factorization over pre-trained single task networks [18], placing a tensor normal prior over the decoder [17] and utilizing a maximum a posteriori optimization objective, or constraining the decoder layers based on the task relations [16]. However, the previous methods suffer from either requiring initially training single task networks [18], modifying the optimization loop [17], or limited to two tasks [16].

Lastly, graphs have seen recent usage in the MTL and MTC fields in modeling between- and within-task relationships. An example of this is the PSD-Net which utilized graphlets to improve per-pixel predictions [12]. For multi-task classification, graph neural networks (GNNs) have been used to model the relationship between the multiple inputs in a batch [13, 14], or across sequential data [15]. In concurrent work [74] a Laplacian graph across facial attributes is learned and used within a regularization term during optimization.

Overall, the literature on MTC decoder-focused models is scarce and existing methods either rely on compressing single task networks or constrained to two tasks. Here, we present a novel decoder-focused model, CT-GNN, which is end-to-end trainable for any number of tasks. Furthermore, in contrast to previous usage of graphs in MTC, the CT-GNN is trainable without relying on sequential or batched data for the graph construction.

3 Methodology

In this section, we present our proposed Cross-Task GNN Decoder for Multi-Task Classification. First we provide a recap of Multi-Task Learning and Graph Neural Networks, followed by an explanation of the CT-GNN decoder and how the graph adjacency matrix can be constructed in a data-driven manner.

3.1 Multi-Task Learning Recap

Multi-Task Learning focuses on the problem of classifying a set of T tasks, $\mathcal{T}$, simultaneously. Each task contains a set of C_t classes, for a total of $C = \sum_t C_t$ classes. In the case of sewer inspection each image, I, has T task-specific labels $\mathbf{y}_t$. The MTL networks are optimized using a linear combination of the task-specific losses:

$$\mathcal{L}_{\text{Total}} = \sum_{t=1}^{T} \lambda_t \mathcal{L}_t(I, \mathbf{y}_t), \tag{D.1}$$

where λ_t and $\mathcal{L}_t$ are the weight and loss of the tth task, respectively.

When applying multi-task learning methods there are typically varying degrees of parameter sharing in the encoder and no parameter sharing in the decoder. An input image is processed by an encoder network, f_{ENC}, and a set of per-task features $\mathbf{x}_t \in \mathbb{R}^{d_{\text{ENC}}}$ are extracted. If there are no task-specific parameters in f_{ENC} all T tasks will use the same encoded feature $\mathbf{x} \in \mathbb{R}^{d_{\text{ENC}}}$. The encoder features are processed by a decoder network, f_{DEC}, producing predictions for each of the tasks, $\tilde{\mathbf{y}}_t \in \mathbb{R}^{C_t}$. Classically, f_{DEC} is constructed as T disjointed classifiers.

3.2 Graph Neural Network Recap

A graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is defined as a set of nodes, $\mathcal{V}$, and edges connecting two nodes, $\mathcal{E}$, together with a set of d-dimensional node features $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$. A graph can be represented using an adjacency matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, where entry $\mathbf{A}[u,v]$ is the edge weight from node v to u. The basic GNN is defined by its neural message passing structure where the feature vectors of the nodes are exchanged and updated, constituting a GNN layer [75]. The neural message passing structure for node u and its neighbors $\mathcal{N}(u)$ is defined as:

$$\mathbf{h}_u^{(l+1)} = \psi(\mathbf{h}_u^{(l)}, \phi(\{\mathbf{h}_v^{(l)}, \forall v \in \mathcal{N}(u)\})), \tag{D.2}$$

where ψ and ϕ are arbitrary differentiable *update* and *aggregation* functions, respectively, and $\mathbf{h}_u^l$ is the hidden embedding of node u at layer l with $\mathbf{h}_u^0 = \mathbf{x}_u$.

3.3 CT-GNN Decoder for Multi-Task Classification

The Cross-Task GNN Decoder builds upon the encoder features, $\mathbf{x}_t$, and consists of the following four parts illustrated in Figure D.2: T task-specific decoder heads producing the initial per-task feature representations, T bottleneck layers reducing the dimensionality of the per-task feature vectors, C non-linear node embedding layers, and a cross-task GNN which jointly refines the different class representations based on an a priori or learned directed graph $\mathcal{G}_{\mathcal{T}}$.

Task-Specific Decoders. The task-specific decoder heads are realized as a set of T disjointed networks, $f_{\text{DEC},t}$, each generating a task-specific feature vectors $\mathbf{z}_t = f_{\text{DEC},t}(\mathbf{x}_t)$, $\mathbf{z}_t \in \mathbb{R}^{d_{\text{DEC}}}$. Classically, $\mathbf{z}_t$ is used directly to obtain the class predictions, $\check{\mathbf{y}}_t$, by applying a linear layer followed by the classification activation function of choice. In the CT-GNN decoder framework, however, the task-feature $\mathbf{z}_t$ is used as the foundation for the class-specific node embeddings, in order to allow for initial task-adaption of the encoder feature, $\mathbf{x}_t$.

Bottleneck Layer. In previous work, the dimensionality of the task-specific feature representation $\mathbf{z}_t$ is equal to that of the encoder feature, meaning $d_{\text{ENC}} = d_{\text{DEC}}$ [56, 64]. In the CT-GNN decoder framework this is problematic, as the model parameter count would increase dramatically when transforming the T task-specific features into C unique class-specific features of size d_{EMB}. Therefore, a non-linear down projection layer, $f_{\text{BTL},t}$, is applied in order to reduce the dimensionality of the task-specific features and generate a more compact feature representation, $\tilde{\mathbf{z}}_t \in \mathbb{R}^{d_{\text{BTL}}}$. The bottleneck is realized as a dense layer, $\tilde{\mathbf{z}}_t = f_{\text{BTL},t}(\mathbf{z}_t) = \sigma(\mathbf{z}_t\mathbf{B}_t)$, consisting of the down projection weight matrix, $\mathbf{B}_t \in \mathbb{R}^{d_{\text{DEC}} \times d_{\text{BTL}}}$, where $d_{\text{BTL}} \leq d_{\text{EMB}} \leq d_{\text{DEC}}$, and applying a differentiable non-linear function, σ. $\mathbf{B}_t$ can be task-specific or shared across all T tasks, depending on the number of tasks. For a large number of tasks, using task-specific bottleneck layers would result in a large parameter increase, decreasing the parameter-wise benefits of using a MTL network.

Node Embeddings. The dimensionality-reduced task feature representation, $\tilde{\mathbf{z}}_t$, is subsequently turned into C_t class-specific node embeddings. $\bar{\mathbf{z}}_{t,c} \in \mathbb{R}^{d_{\text{EMB}}}$. Similar to the bottleneck layer, this is realized as a dense layer, $\bar{\mathbf{z}}_{t,c} = f_{\text{EMB},t,c}(\tilde{\mathbf{z}}_t) = \sigma(\tilde{\mathbf{z}}_t\mathbf{E}_{t,c})$, consisting of a matrix multiplication and non-linearity. In order to get the C_t unique node embeddings, we use C_t unique embedding layers, parameterized by C_t unique learnable matrices $\mathbf{E}_{t,c} \in \mathbb{R}^{d_{\text{BTL}} \times d_{\text{EMB}}}$.

Cross-Task GNN. The stacked initial per-class node embeddings, $\bar{\mathbf{Z}} \in \mathbb{R}^{C \times d_{\text{EMB}}}$, of the cross-task graph, $\mathcal{G}_{\mathcal{T}}$, are refined by passing them through a GNN, $\hat{\mathbf{Z}} = f_{\text{GNN}}(\bar{\mathbf{Z}})$, where $\hat{\mathbf{Z}} \in \mathbb{R}^{C \times d_{\text{EMB}}}$ is the stacked GNN-refined node features. The GNN fundamentally builds upon an adjacency matrix of $\mathcal{G}_{\mathcal{T}}$, $\mathcal{A} \in \mathbb{R}^{C \times C}$, which can be learned, provided a priori, or obtained by a combination thereof. The GNN propagates the node embeddings through L hidden layers with d_{EMB} channels, adding contextual information to each node embedding based on its incoming neighbors.

Each node embedding, $\hat{\mathbf{z}}_{t,c} \in \mathbb{R}^{d_{\text{EMB}}}$, is passed through a class-specific linear projection layer, $\hat{z}_{t,c} = f_{\text{CLS},t,c}(\hat{\mathbf{z}}_{t,c})$, to generate a scalar node embedding for each class. The scalar embeddings, $\hat{z}_{t,c}$, are stacked per-task, and the task-specific activation functions are applied to generate the per-task probability vectors, $\hat{\mathbf{y}}_t$. For multi-label and multi-class classification we use the sigmoid and softmax activation.

3.4 Adjacency Matrix Construction

A key part of the CT-GNN Decoder is the construction of the graph, realized by the adjacency matrix $\mathcal{A}$. This adjacency matrix can in theory be arbitrarily set. However, in order to utilize the a priori knowledge of the task relationships, we follow a data-driven approach based on the co-occurrence of the classes. We generalize the graph construction method Chen *et al.* [76] to the multi-task classification scenario.

$\mathcal{A}$ consists of several sub-matrices, $\mathcal{A}_{i,j}$, each describing the relationship between the tasks i and j. Note that in the case that only binary and multi-class classification tasks are considered, $\mathcal{A}$ will be a directed T-partite graph with self-loops. Firstly, the conditional probabilities between the classes in task i and j, $\mathbf{P}_{i,j} \in \mathbb{R}^{C_i \times C_j}$, are calculated based on the co-occurrence matrix between the two tasks, $\mathbf{C}_{i,j} \in \mathbb{R}^{C_i \times C_j}$, see Eq. D.3–D.4. The co-occurrence matrices are calculated using the training splits. We follow the convention that $\mathbf{P}_{i,j}[u,v]$ defines the conditional probability of class u given class v.

$$\mathbf{P}_{i,j}[u,v] = \frac{\mathbf{C}_{i,j}[u,v]}{N_v} \tag{D.3}$$

$$N_v = \begin{cases} \mathbf{C}_{i,j}[v,v], & i = j \\ \sum_{u=1}^{C_i} \mathbf{C}_{i,j}[u,v], & i \neq j \end{cases} \tag{D.4}$$

$\mathbf{P}_{i,j}$ is subsequently binarized in order to filter out noisy edges using a task-pair specific threshold $\tau_{i,j}$, see Eq. D.5. By utilizing task-pair specific thresholds the different task-pairs can be binarized according to different rules, if desired. The binarized adjacency matrices are then combined into a single adjacency matrix, $\mathbf{A}$, see Eq. D.6.

$$\mathbf{A}_{i,j}[u,v] = \begin{cases} 0, & \mathbf{P}_{i,j}[u,v] < \tau_{i,j} \\ 1, & \mathbf{P}_{i,j}[u,v] \geq \tau_{i,j} \end{cases} \tag{D.5}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \dots & \mathbf{A}_{1,K} \\ \vdots & \ddots & \vdots \\ \mathbf{A}_{K,1} & \dots & \mathbf{A}_{K,K} \end{bmatrix} \tag{D.6}$$

Lastly, the adjacency matrix is re-weighted across the incoming edges per node, in order to counteract the oversmoothing problem with GNNs [76], leading to the final adjacency matrix, $\mathcal{A}$. This is done using $\mathbf{A}$, and enforcing the sum of all incoming edge weights to equal one, setting the sum of the neighbor edge weights to p, while the center node self-loop weight is $1-p$, see Eq. D.7.

$$\mathcal{A}[u,v] = \begin{cases} \mathbf{A}[u,v] \frac{p}{\sum_{v=1,v\neq u}^{C_j} \mathbf{A}[u,v]}, & u \neq v \\ 1-p, & u = v \end{cases} \tag{D.7}$$

The larger p is the more weight will be assigned to the incoming neighbor nodes, while a smaller p value will result in more weight assigned to the center node. If a center node has no incoming edges a part from the self-loop, *i.e.* $\sum_{v=1,v\neq u}^{C_j} \mathbf{A}[u,v] = 0$, we set the self-loop weight to one, to avoid the center node embedding decaying to a zero vector.

4 Experimental Results

We evaluate on the Sewer-ML sewer defect and pipe property dataset [1]. The dataset focuses on the multi-label defect classification problem and contains 1.3 million images collected over a nine year period. The data are split into a preset training, validation, and test split, containing 1 million, 130k, and 130k images each [1]. The defect classification problem consists of 17 different classes as well as the implicit normal class. Additionally, the water level, pipe material and pipe shape are also annotated. The water level is annotated in 11 classes from 0 to 100% of the pipe filled with water in 10% steps, and the pipe material and shape tasks contain eight and six classes each. Example images can be found in the supplementary material.

4.1 Evaluation Metrics

Model evaluation is done using the per-task evaluation metrics and number of parameters, #P. As the classes in each task are imbalanced the tasks cannot be evaluated using the traditional accuracy metric. Instead, the defect task is evaluated using the $\mathrm{F2}_{\mathrm{CIW}}$ defect score and the $\mathrm{F1}_{\mathrm{Normal}}$ score [1]. The three remaining tasks are evaluated using both the micro-F1 (mF1) and macro-F1 (MF1) scores.

Lastly, we report the average per-task performance increase for a multi-task model, Δ_{MTL}, with respect to the single task learning (STL) baselines of the same base architecture [77]:

$$\Delta_{\mathrm{MTL}} = \frac{1}{T}\sum_{t=1}^{T}\frac{(M_{m,t} - M_{b,t})}{M_{b,t}}, \tag{D.8}$$

where $M_{m,t}$ and $M_{b,t}$ are the multi-task and single-task metric performance for task t, receptively.

4.2 Training Procedure

We utilize the ResNet-50 network [78] as our base encoder, with no task-specific decoders, meaning $\mathbf{x}_t = \mathbf{z}_t$. We cast the defect classification problem as a multi-label classification task with a single task weight, $\lambda_{\mathrm{defect}}$, while the water level, pipe material, and pipe shape are multi-class classification tasks. For the water level classification task, we adapt the label discretization approach from [44], leading to four water level classes.

We compare performance using the Graph Convolutional Network (GCN) [79] and Graph Attention Network (GAT) [80] in the CT-GNN, denoting the variations CT-GCN and CT-GAT, respectively. We use the reweighted adjacency matrix, $\mathcal{A}$, for GCN, and the binary adjacency matrix, $\mathbf{A}$, for GAT where the edge weights are inferred through self-attention. While the GAT architecture could fully determine the adjacency matrix through self-attention, we found that performance increases if we provide the set of possible graph edges beforehand. The GCN adjacency matrix was symmetrically normalized [79] using the in-degree matrix, and skip connections were inserted between the GNN layers. Finally, we use task-specific bottleneck layers.

Hyperparameters. The networks are trained for 40 epochs using SGD with a learning rate of 0.1, momentum of 0.9, weight decay of 0.0001, and a batch size of 256. The learning rate is multiplied by 0.01 at the 20th and 30th epoch. The hyperparameters used in the CT-GNN, including the number of attention heads in GAT, H, are described in Table D.3, and are found through a sequential hyperparameter search described in the supplementary material. Through initial tests we found that a single global threshold τ in the adjacency graph construction leads to the best performance.

Table D.3: CT-GNN hyperparameters. The hyperparameters were found through a sequential search. L is the number of layers in the CT-GNN, d_{ENB} is the dimensionality of the class features, d_{BTL} is the dimensionality of the bottleneck, H is the number of attention heads in the GAT GNN, and τ and p are the thresholding and re-weighting parameters in the adjacency matrix construction, respectively.

Hyperparameter	L	d_{EMB}	d_{BTL}	H	τ	p
GCN	3	512	32	-	0.05	0.2
GAT	1	128	32	8	0.65	-

Data Augmentation. We follow the data augmentation process by [1], rescaling the images to 224 × 224, horizontal flipping and jittering the brightness, contrast, hue, and saturation values by ±10%. Due to class imbalance in each task, we use class-weighted task-losses with the class weighting method of [81] with $\beta = 0.9999$, except for the defect task where the positive class examples are weighted by their *class importance weights* (CIW) [1].

Loss considerations. For all CT-GNN models the final task loss is a convex combination of the final probability vector $\hat{\mathbf{y}}_t$ and the probability vector produced by applying a classification layer to $\mathbf{z_t}$, denoted $\check{\mathbf{y}}_t$:

$$\mathcal{L}_t = \omega \mathcal{L}_t(\hat{\mathbf{y}}_t, \mathbf{y}_t) + (1-\omega)\mathcal{L}_t(\check{\mathbf{y}}_t, \mathbf{y}_t), \tag{D.9}$$

where $\mathcal{L}_t$ is the task-specific loss function for task t, and ω is a weighting hyperparameter in the interval $[0,1]$. This is to ensure the feature representation $\mathbf{z_t}$ is representative for task t, through an auxiliary loss signal. We set $\omega = 0.75$, such that the primary loss signal is propagated through the CT-GNN.

We constrain the task weights to be a convex combination and set to $\lambda_{\text{defect}} = 0.90$ and $\lambda_{\text{water}} = \lambda_{\text{shape}} = \lambda_{\text{material}} = \frac{1-\lambda_{\text{defect}}}{3}$. In order to keep the losses comparable across different settings, we multiply the task weights by T such that $\sum_t \lambda_t = T$, similar to [57].

4.3 Comparative Models

As there are no ResNet-50 STL baselines for all of the tasks, we train these using the same hyperparameters as the in MTL networks. Note that we got the best single-task performance for the defect task using the class weighting method from [81]. We also compare with the benchmark defect classification model from [1], as well as the water level classification model from [44]. As there are no prior work on multi-task classification in the sewer domain [4], we compare with a set of MTL baselines: A hard-shared ResNet-50 MTL network with no CT-GNN (R50-MTL), and the encoder-focused soft-shared MTAN model with a ResNet-50 backbone, see Table D.4. Results for the DWA [57] and the uncertainty [58, 59] optimization-based methods can be found in the supplementary materials.

Table D.4: Results on Sewer-ML. Comparison between the STL and MTL networks. We compare the effect of CT-GNN using GCN and GAT, denoted CT-GCN and CT-GAT respectively, as well as compare a hard-shared ResNet-50 encoder, and the soft-shared MTAN encoder with a ResNet-50 backbone. #P indicates the number of parameters in millions. * indicates that the method was tested on a subset of the Sewer-ML dataset. Best performance in each column is denoted in **bold**.

	Model		Overall	Defect		Water		Shape		Material	
	Model	#P	Δ_{MTL}	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**	**MF1**	**mF1**	**MF1**	**mF1**	**MF1**	**mF1**
Validation Split	Benchmark [1]	62.8	-	55.36	91.32	-	-	-	-	-	-
	R50-FT* [44]	23.5	-	-	-	62.53	78.15	-	-	-	-
	STL	94.0	+0.00	58.42	**92.42**	69.11	79.71	46.55	98.06	65.99	96.71
	R50-MTL	23.5	+10.36	59.73	91.87	70.51	80.47	71.64	99.34	80.28	98.09
	MTAN	48.2	+10.40	61.21	92.10	70.06	**80.59**	68.34	**99.40**	83.48	**98.25**
	CT-GCN	25.2	+12.39	61.35	91.84	**70.57**	80.47	**76.17**	99.33	82.63	98.18
	CT-GAT	24.0	+**12.81**	**61.70**	91.94	**70.57**	80.43	74.53	**99.40**	**86.63**	98.24
Test Split	Benchmark [1]	62.8	-	55.11	90.94	-	-	-	-	-	-
	R50-FT* [44]	23.5	-	-	-	62.88	79.29	-	-	-	-
	STL	94.0	+0.00	57.48	**92.16**	69.87	80.09	56.15	97.59	69.02	96.67
	R50-MTL	23.5	+7.39	58.29	91.57	71.17	81.09	79.48	99.19	**76.35**	98.08
	MTAN	48.2	+6.83	59.91	91.72	70.61	**81.16**	78.50	99.21	72.73	**98.27**
	CT-GCN	25.2	+7.64	60.07	91.60	70.69	80.91	80.32	99.19	75.13	98.15
	CT-GAT	24.0	+**7.84**	**60.57**	91.61	**71.30**	80.91	**81.10**	**99.22**	73.95	98.26

4.4 Results

We find that the CT-GNN outperforms all other methods, beating state-of-the-art defect [1] and water level [44] classifiers by 5.3 and 8.0 percentage points, respectively. We also outperform the baseline STL and MTL networks, by a significant margin on the defect, shape, and material tasks.

The CT-GCN and CT-GAT achieve comparable or better metric performance on all tasks while adding 0.5-1.7 million parameters compared to MTAN encoder-focused method which adds 25 million parameters. Specifically, CT-GAT achieves the highest Δ_{MTL} while introducing 50 times fewer parameters than the MTAN encoder. Unlike soft-shared encoders, the backbone only influences the parameter count of the CT-GNN through the size of the encoder feature $\mathbf{x}_t$.

Comparatively, the optimization-based methods performed worse than using a fixed set of task weights, echoing the results from [5], resulting in a Δ_{MTL} of -15.70% and -4.07% on the validation split and -11.57% and -4.07% on the test split, for the DWA and uncertainty methods respectively. Details are available in the supplementary material.

We also find that the CT-GAT outperforms the CT-GCN on the defect and materials task, while the CT-GCN performs slightly better on the shape task MF1 score. This indicates that there is a clear value in letting the edge weights be dynamically inferred during inference, while prior information can be imbued beforehand through the structure of the adjacency matrix. Furthermore, it demonstrates that good performance can be achieved with limited prior knowledge of the task and class relationships.

Lastly, we observe that the general performance, as measured by Δ_{MTL}, increases when using MTL networks. By inspecting the results, one can see that the water task performance is not affected by the MTL networks. However, for the defect, material and shape tasks the performance increases dramatically, beating the STL method and benchmark method from [1] by several percentage points, indicating a clear benefit of utilizing an MTL approach. We also observe a clear difference in Δ_{MTL} across the validation and test splits. This is attributed to the shape and material tasks where the classes are very imbalanced, leading to few labels to learn from during training and a potentially large difference between the examples in the different splits.

4.5 Per-task Analysis of Results

To get a better understanding of the performance difference between the CT-GNNs and R50-MTL, we dive into the per-class task performances. Images of the different classes can be found in the supplementary material.

When comparing the individual defect F2-scores, shown in Figure D.4a, we see that the CT-GNN performs better on defects with high CIWs but few training examples such as OS, PB, and PS, while the performance is worse on the rare defect classes with a low CIW such as IS and FO. For classes where there are plenty of examples to learn from we observe that the performance is comparable across all models.

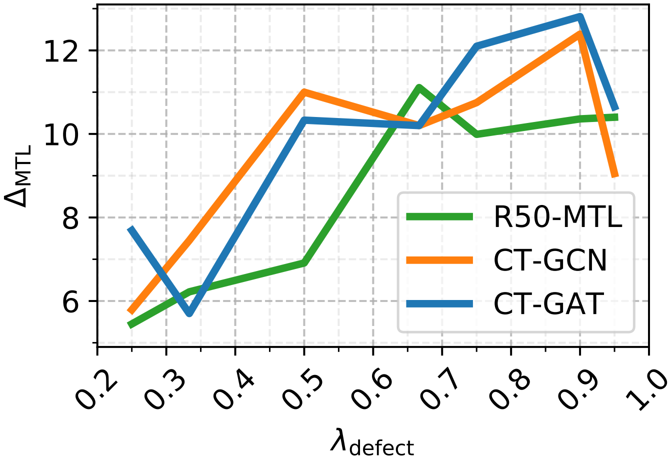


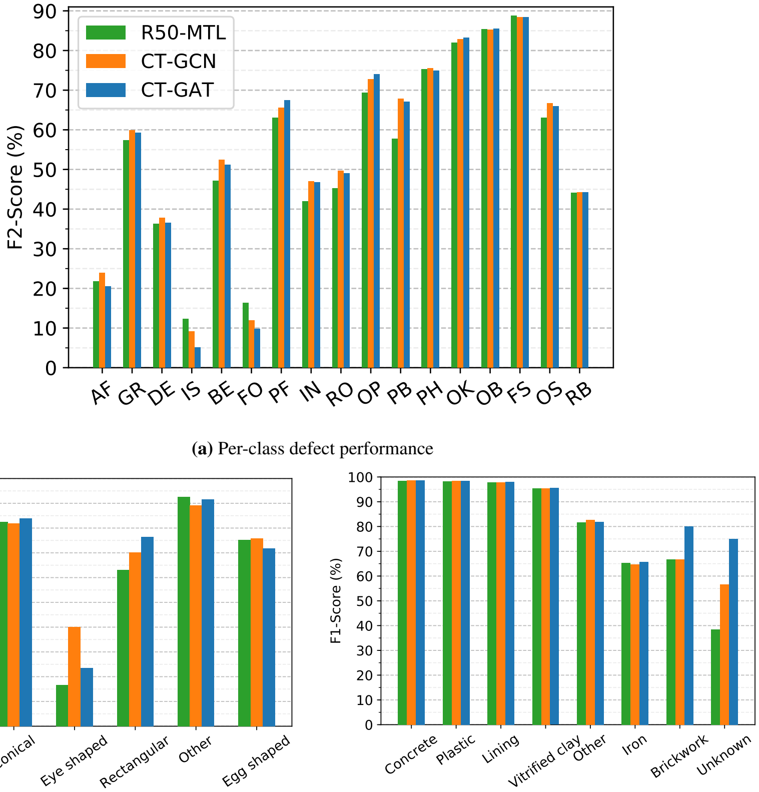
Fig. D.3: Evaluating Δ_{MTL} for different $\lambda_{\mathrm{defect}}$. Comparison of performance of the R50-MTL, CT-GCN and CT-GAT models. Evaluated on the validation split.

When investigating the water task, we observe that all models perform equally well on all classes. On the shape task it is clear the CT-GNN performs better on the rectangular and eye shaped pipes, see Figure D.4b. It should be noted that the amount of validation examples of eye shaped pipes is very low. The CT-GNN does, however, achieve a slightly lower F1-score on the egg shaped pipes. On the material task, the CT-GNN again improves performance compared to the baseline, see Figure D.4c. By using the CT-GAT performance on the Brickwork and Unknown classes increase by 13 and 37 percentage points, respectively.

4.6 Ablation Studies

Importance of $\lambda_{\mathrm{defect}}$. The most critical part of an automated sewer inspection system, is the capability to classify the presence of defects correctly. Therefore, we investigate the effect of different $\lambda_{\mathrm{defect}}$ values on the overall performance metric Δ_{MTL}. We compare the performance when setting $\lambda_{\mathrm{defect}} = \{0.25, 0.33, 0.50, 0.67, 0.75, 0.90, 0.95\}$ ranging from an equal weighting between all four tasks ($\lambda_{\mathrm{defect}} = 0.25$) to focusing on the defect task ($\lambda_{\mathrm{defect}} = 0.95$). We train an MTL model with a hard-shared ResNet-50 encoder with and without the CT-GNN decoder heads, see Figure D.3. We observe that the Δ_{MTL} increases steadily together with $\lambda_{\mathrm{defect}}$, peaking at $\lambda_{\mathrm{defect}} = 0.90$, before decreasing when prioritizing the defect task too much when $\lambda_{\mathrm{defect}} = 0.95$.

Combining MTAN and CT-GNN. The combination of soft parameter sharing encoder- and decoder-focused models has not previously been investigated. Therefore, we compare the effect of combining MTAN encoder and the CT-GNN decoder, to determine whether the two approaches are complementary. We find that the CT-



(a) Per-class defect performance

(b) Per-class shape performance

(c) Per-class material performance

Fig. D.4: Per task class comparisons. We compare model performance on the validation set. The F2 defect scores are plotted for each defect class in Figure D.4a ordered by increasing CIW from left to right. We refer to the Sewer-ML paper [1] for an explanation of the defect class codes. The class F1-scores for the shape and material tasks are plotted in Figure D.4b-D.4c. The scores are plotted by decreasing number of training samples per class.

GCN and CT-GAT obtains a Δ_{MTL} of 12.72% and 11.48% when trained with MTAN, respectively. This shows that the combination of MTAN and CT-GCN leads to a higher performance with the CT-GCN compared to using a hard-shared encoder. However, when using the CT-GAT the performance decreases. This indicates the GNN settings cannot just be transferred from a hard to soft-shared encoder, instead requiring a small search over how the graph is constructed.

The per-task metric performances for both ablation studies can be found in the supplementary material. h

5 Conclusion

One of the most important infrastructures in modern society is the sewerage infrastructure, but it is difficult to inspect and maintain. Automated sewer inspection methods have been investigated for decades, with an emphasis of the important defect classification task, while sewer properties such as water level, pipe material, and pipe shape, which are needed to determine the deterioration level, have been neglected.

We approach the automated sewer inspection problem as a multi-task classification problem. To this end we introduce our novel Cross-Task Graph Neural Network (CT-GNN) Decoder, which utilizes the cross-task information between concurrent and related tasks to refine the per-task predictions. This is realized by generating unique per-class node embeddings that are combined and refined through the use of a graph neural network.

Using our novel method, we not only beat the state-of-the-art on the defect and water level classification tasks by 5.3 and 8.0 percentage points, respectively, but also outperform other single-task and multi-task learning methods on all four classification tasks in the Sewer-ML dataset [1]. Furthermore, the CT-GNN decoder introduces 50 times fewer parameters compared to encoder-focused models.

The novel CT-GNN approach is focused on handling the concurrent image-level classification tasks present in the Sewer-ML dataset. It is, however, important to note that the method is not specific to the sewer data and can therefore be expected to generalize to other domains containing concurrent classification tasks. Another interesting future direction for the CT-GNN is to adapt it to regression tasks where the values cannot be discretized.

Acknowledgments

This research was funded by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project, and partially supported by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE), and by ICREA under the ICREA Academia programme.

D.A Supplementary Materials Content

In these supplementary materials we describe the hyperparameter search, more in-depth results for the optimization-based multi-task learning (MTL) methods as well as the ablation studies. We also show examples of the different task classes, and show examples of success and failure cases for the CT-GNN. Specifically, the following will be described:

- Example images of the different task classes (Section D.B).
- Hyperparameter search (Section D.C).
- In-depth optimization-based MTL results (Section D.D).
- In-depth results for the λ_{defect} ablation study (Section D.E).
- In-depth results for the MTAN and CT-GNN ablation study (Section D.F).
- Examples of how the CT-GNN succeeds and fails (Section D.G).

D.B Sewer-ML Task Class Examples

For the sake of clarity we show examples of each class in the water level, pipe shape and pipe material tasks, see Figure D.14-D.16. For examples of the pipe defect classes we refer to the supplementary materials of the Sewer-ML paper [1].

D.C Hyperparameter Search

In the hyperparameter search for the CT-GNN decoder we investigated the effect when varying the design of the bottleneck layer and the CT-GNN. The investigated parameters and their search space is presented in Table D.6. It should be noted that the amount of attention heads, H, and the re-weighting parameter, p, were only utilized for the GAT [80] and GCN [79] GNNs, respectively. Due to the amount of hyperparameters and the size of the value ranges, we decided to employ a sequential hyperparameter search design. The search was initialized with the hyperparameters stated in Table D.5. All tests were performed with $\lambda_{\text{defect}} = 0.50$ to ensure a fair weighting of the tasks, while prioritizing the defect task.

At each step of the search the best performing hyperparameter was kept and used for all future steps of the search. The order of the sequential search was realized as follows:

1. Grid search across the number of GNN layers, L, and the number of GNN channels, d_{EMB}.
2. Search over the number of channels in the bottleneck layer, d_{BTL}.
3. Search over the number of attention heads, H. **Only performed for GAT.**

Table D.5: Initial Hyperparameter Values. The investigated hyperparameters are set to the following starting values, and after each step of the sequential search the corresponding hyperparameter is updated. It should be noted that τ used in the GAT GNN was set to 0.05. This was done to reduce the amount of noisy graph edges in the Sewer-ML dataset, caused by the large class imbalance in some tasks.

Hyperparameter	GCN	GAT
L	2	2
d_{EMB}	256	256
d_{BTL}	32	32
H	-	8
τ	0.05	0.05
p	0.2	-

Table D.6: Investigated Hyperparameters. The hyperparameters of the CT-GNN and the Bottleneck layer were investigated. For each hyperparameter we have denoted the values investigated.

Hyperparameter	Range
L	[1, 2, 3]
d_{EMB}	[128, 256, 512]
d_{BTL}	[16, 32, 64, 128]
H	[1, 2, 4, 8, 16]
τ	[0.00, 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95]
p	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

4. Search over the adjacency matrix threshold, τ.
5. Search over the adjacency matrix neighbor node reweighting parameter, p. **Only performed for GCN.**

The results of the sequential hyperparameter search on the Sewer-ML dataset are shown in Figures D.5-D.10. From these results we can conclude that the performance when using the GAT leads to more stable performances as the Δ_{MTL} in general does not vary as wildly as when using the GCN. However, when using the GCN we achieve in general higher Δ_{MTL}. We can also observe that the adjacency matrix threshold τ has a large effect on the performance. Specifically, it is observable that using a low τ of 0.05 leads to good performance, which is only matched when τ is set to 0.65 and above for the GAT and 0.35 and above for the GCN. Lastly, we observe that an increased neighbor node reweighting parameter p leads to degraded performance, indicating that the center-node information is crucial. The conditional probability matrix, the binary matrices with τ set to 0.05 and 0.65, as well as the reweighted adjacency matrix with $\tau = 0.05$ and $p = 0.2$ are shown in Figure D.11a-D.11d.

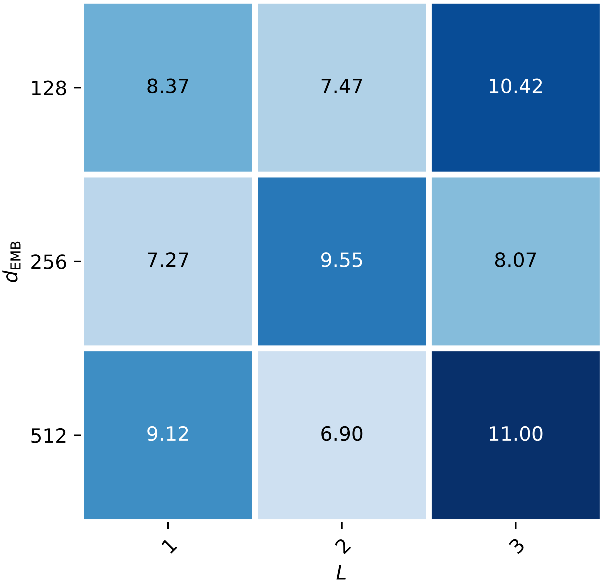


Fig. D.5: Grid search over L and d_{EMB} for CT-GCN.

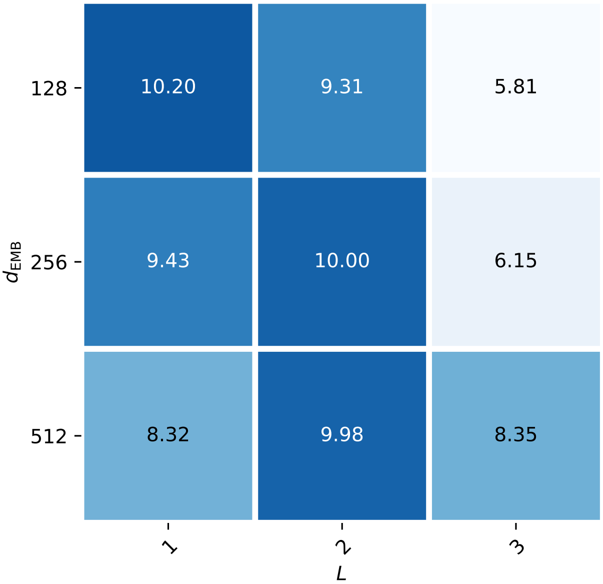


Fig. D.6: Grid search over L and d_{EMB} for CT-GAT.

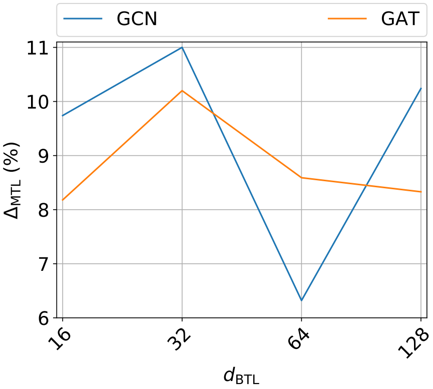


Fig. D.7: Search over d_{BTL}.

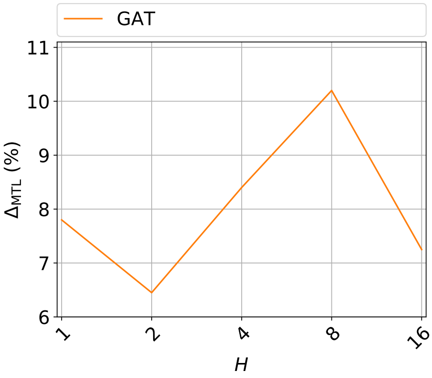


Fig. D.8: Search over H.

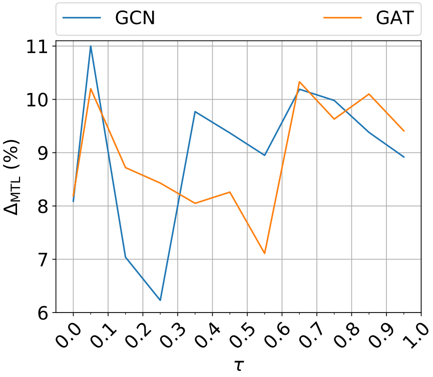


Fig. D.9: Search over τ.

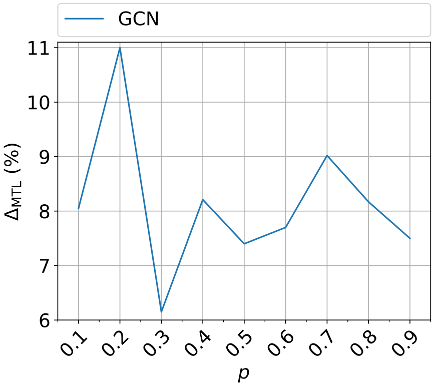


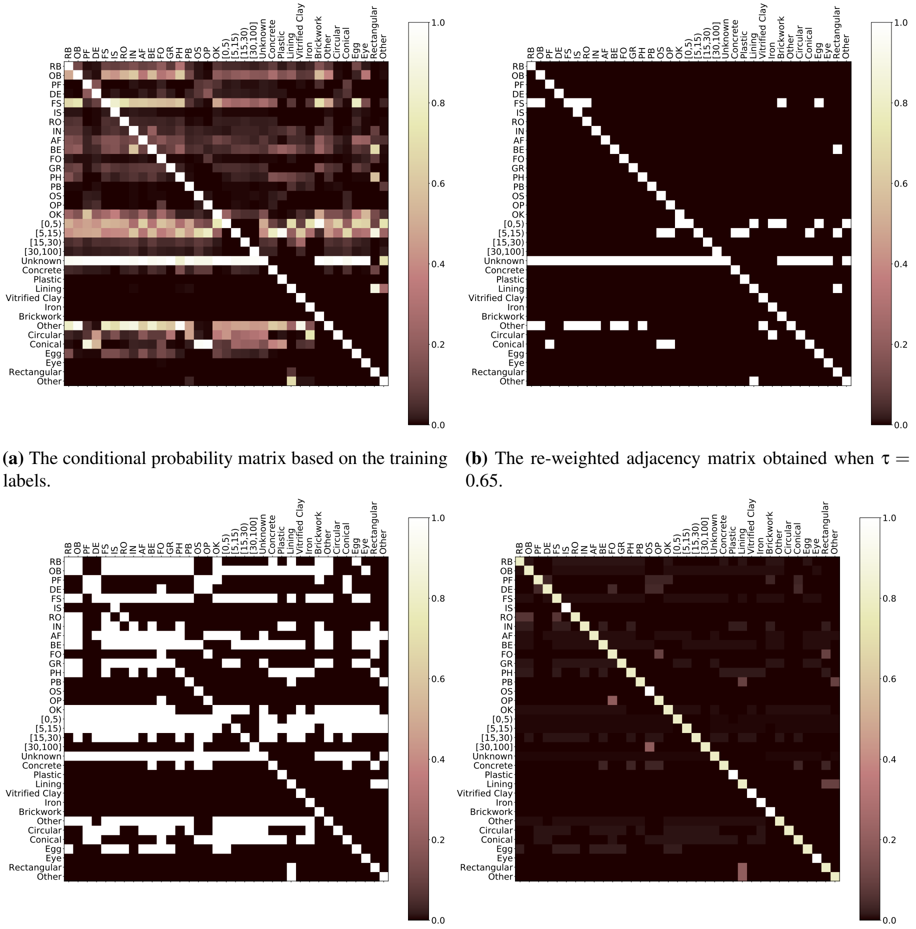
Fig. D.10: Search over p.

D.D Optimization-Based MTL - In-Depth Results

We present the full results for the optimization-based method Dynamic Weight Averaging (DWA) [57] and Uncertainty estimation (Uncrt.) [58, 59], see Table D.7. The DWA task weighting method is initialized with $\lambda_{\mathrm{defect}} = 0.90$, while Uncrt. is initialized with unit variances for each task. From the results we observe that the DWA method performs worse than the STL networks on nearly every task. The Uncrt. method improves the shape and material MF1 compared to the STL networks, but suffers from poor defect classification rate.

D.E Effect of $\lambda_{\mathrm{defect}}$ - In-Depth Results

We show the in-depth results for each tested setting of $\lambda_{\mathrm{defect}}$ on the validation split for the R50-MTL baseline as well as CT-GNNs, see Table D.9. We observe that a larger



(a) The conditional probability matrix based on the training labels.

(b) The re-weighted adjacency matrix obtained when $\tau = 0.65$.

(c) The re-weighted adjacency matrix obtained when $\tau = 0.05$.

(d) The re-weighted adjacency matrix when $\tau = 0.05$ and $p = 0.2$.

Fig. D.11: Adjacency matrix construction. We show the conditional probability matrix across task classes, as well as the constructed binary and reweighted adjacency matrices.

λ_{defect} leads to a higher Δ_{MTL} due to a higher F2_{CIW}. However, it also leads to a lower material MF1 score, as we observe that the material MF1 score peaks at 90.5% for the CT-GNNs when $\lambda_{\text{defect}} = 0.50$, and decreases to 82-86% when $\lambda_{\text{defect}} = 0.90$.

D.F Combining the MTAN Encoder and CT-GNN Decoder - In-Depth Results

We present the in-depth results of the ablation studies investigating the combination of MTAN encoder and the CT-GNN Decoder, see Table D.8. The methods were only evaluated on the validation split. From the results we see that the Δ_{MTL} is increased by introducing the CT-GNN, and that the combination with the CT-GCN outperforms using a hard-shared encoder. We observe that the noticeable difference is in the defect classification task where the performance is increased by 0.6-0.7 percentage points on the F2_{CIW} metric.

D.G CT-GNN Success and Failure Cases

We show several cases where the CT-GNN decoder correctly classifies all tasks, shown in Figure D.12, as well as cases where some or all tasks are misclassified, shown in Figure D.13.

We observe that the the CT-GNN performs well when several defects occur at the same time at different distances to the camera (see top left example), as well as subtle defects such as the distortion in the bottom middle example and crack in the bottom left example. Similarly, this can be observed in the top right example where the high water level is detected even though it is partially occluded and unlit. Lastly it can correctly handle rare classes such as the iron material in the bottom right example.

In Figure D.13 we observe that the the CT-GNN misclassify irregularities in the pipe geometry as displaced pipes (FS) or construction changes (OK), as seen in the top right and top middle examples. In both cases the predictions is understandable as the internal reparation is shifted (top left) and the camera is placed right before a well (top middle). In the top right case the deformation is observed as a surface damage, which is understandable due to the folds of the deformation. For the cases where all classifications are incorrect, we see that the CT-GNN decoder misclassifies several tasks due to limited context introduced by the camera perspective.

Table D.7: Effect of optimization-based methods. In-depth results for two optimization-based methods, DWA [57] and the uncertainty (Uncrt.) based method [58, 59]. TW indicates the task weighting method used and #P indicates the number of parameters in millions. The best performance in each column is denoted in **bold**.

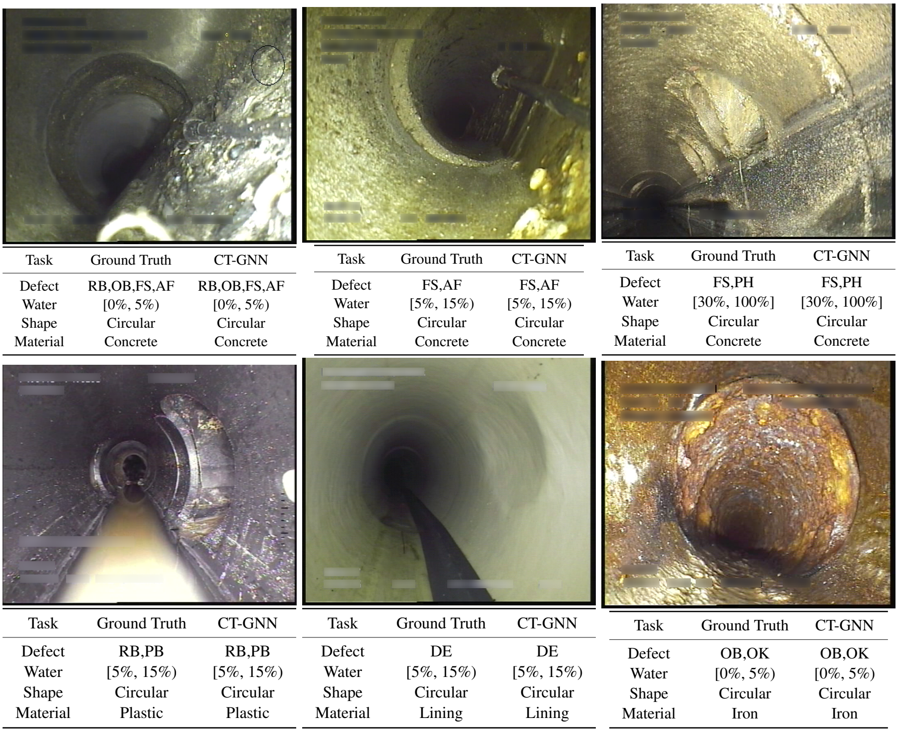
	Model			Overall	Defect		Water		Shape		Material	
	Encoder	TW	#P	Δ_{MTL}	F2_{CIW}	$\text{F1}_{\text{Normal}}$	**MF1**	**mF1**	**MF1**	**mF1**	**MF1**	**mF1**
Val.	STL	-	94.0	**+0.00**	**58.42**	**92.42**	**69.11**	**79.71**	46.55	98.06	65.99	**96.71**
	R50-MTL	DWA	23.5	-15.70	34.22	86.57	53.43	70.83	37.68	98.18	53.50	90.79
	R50-MTL	Uncrt.	23.5	-4.07	24.80	86.80	62.00	75.31	**67.30**	**99.19**	**67.46**	95.66
Test	STL	-	94.0	**+0.00**	**57.48**	**92.16**	**69.87**	**80.09**	56.15	97.59	69.02	**96.67**
	R50-MTL	DWA	23.5	-11.57	34.84	86.20	54.30	71.03	59.27	97.81	60.39	90.49
	R50-MTL	Uncrt.	23.5	-3.78	26.30	86.48	63.01	76.15	**79.69**	**98.99**	**70.84**	95.59

Table D.8: Effect of encoder. We compare the effect of training CT-GNN using GCN and GAT with the MTAN encoder, and with fixed task weights. #P indicates the number of parameters in millions. Evaluated on the validation split. The best performance in each column is denoted in **bold**.

Model			Overall	Defect		Water		Shape		Material	
Encoder	CT-GNN	#P	Δ_{MTL}	F2_{CIW}	$\text{F1}_{\text{Normal}}$	**MF1**	**mF1**	**MF1**	**mF1**	**MF1**	**mF1**
MTAN	✗	48.2	+10.40	61.21	**92.10**	70.06	**80.59**	68.34	99.40	83.48	98.25
MTAN	GCN	49.9	**+12.72**	61.86	91.99	**71.39**	80.53	**75.42**	**99.46**	**83.77**	98.25
MTAN	GAT	48.6	+11.48	**61.92**	92.03	70.95	80.50	71.17	99.39	83.65	**98.29**

Table D.9: Effect of λ_{defect}. We compare the performance of the R50-MTL baseline and CT-GNN heads when training with different λ_{defect} values. Evaluated on the validation split. The best performance in each column is denoted in **bold** per method.

Model		Overall	Defect		Water		Shape		Material	
Model	λ_{defect}	Δ_{MTL}	**F2_{CIW}**	**$\text{F1}_{\text{Normal}}$**	**MF1**	**mF1**	**MF1**	**mF1**	**MF1**	**mF1**
R50-MTL	0.25	+5.45	32.86	88.40	69.42	79.85	74.72	99.21	84.64	97.83
	0.33	+6.22	39.85	89.08	69.18	79.89	70.29	99.31	86.61	97.96
	0.50	+6.91	40.78	89.31	69.35	79.90	71.58	99.25	87.21	97.75
	0.67	**+11.11**	52.53	90.69	70.19	80.22	**75.74**	99.38	**87.83**	98.11
	0.75	+9.99	56.31	91.41	70.15	80.42	69.91	**99.40**	85.13	**98.30**
	0.90	+10.36	59.73	**91.87**	**70.51**	**80.47**	71.64	99.34	80.28	98.09
	0.95	+10.40	**60.34**	91.85	69.35	80.02	71.85	99.19	81.26	97.82
CT-GCN	0.25	+5.79	39.44	88.77	69.76	79.63	73.67	99.27	80.06	97.79
	0.33	+7.45	42.56	89.12	69.36	79.82	72.20	99.20	87.40	97.81
	0.50	+11.00	50.35	90.01	70.04	79.98	75.80	**99.44**	**90.54**	97.96
	0.67	+10.20	54.67	90.64	69.78	79.92	72.94	99.35	85.31	98.06
	0.75	+10.75	57.71	91.11	70.48	80.21	70.95	99.37	86.27	**98.21**
	0.90	**+12.39**	61.35	91.84	**70.57**	**80.47**	**76.17**	99.33	82.63	98.18
	0.95	+9.05	**62.10**	**92.01**	69.95	80.04	67.36	99.11	77.83	97.89
CT-GAT	0.25	+7.69	37.02	88.69	70.06	80.18	75.47	99.45	89.40	97.89
	0.33	+5.70	42.17	89.09	69.54	79.96	71.72	99.37	78.80	97.95
	0.50	+10.33	49.96	89.98	69.69	79.90	73.90	99.41	**90.52**	98.06
	0.67	+10.20	55.26	90.69	69.80	80.38	72.41	99.40	84.90	98.12
	0.75	+12.10	58.37	91.45	70.46	**80.43**	**76.82**	**99.46**	83.75	**98.35**
	0.90	**+12.81**	**61.70**	**91.94**	**70.57**	**80.43**	74.53	99.40	86.63	98.24
	0.95	+10.65	60.95	92.03	69.01	79.59	70.75	99.18	83.99	97.84



Task	Ground Truth	CT-GNN
Defect	RB,OB,FS,AF	RB,OB,FS,AF
Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular
Material	Concrete	Concrete

Task	Ground Truth	CT-GNN
Defect	FS,AF	FS,AF
Water	[5%, 15%)	[5%, 15%)
Shape	Circular	Circular
Material	Concrete	Concrete

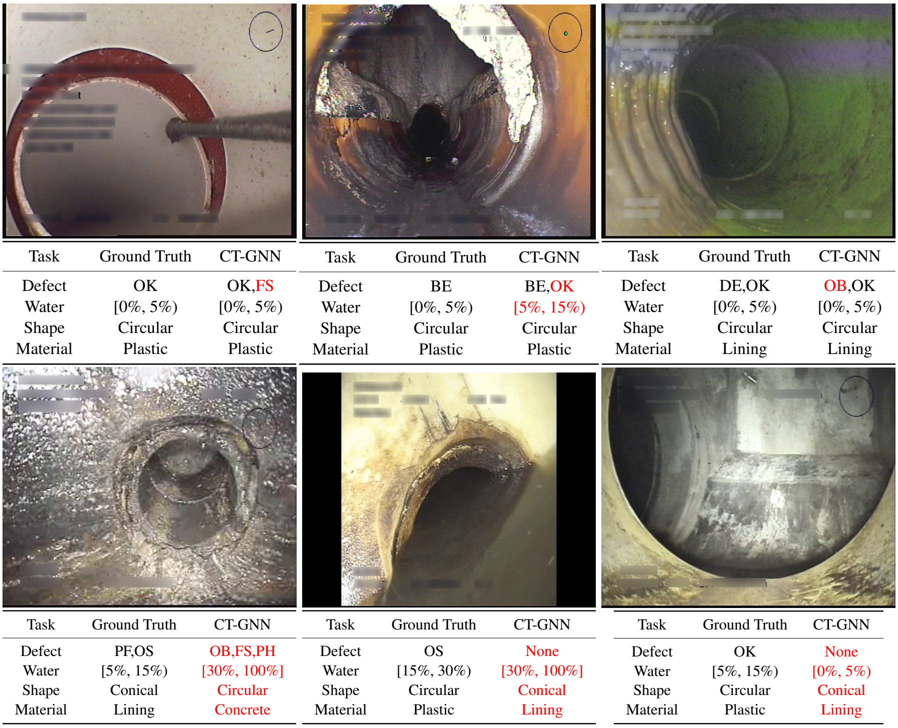
Task	Ground Truth	CT-GNN
Defect	FS,PH	FS,PH
Water	[30%, 100%]	[30%, 100%]
Shape	Circular	Circular
Material	Concrete	Concrete

Task	Ground Truth	CT-GNN
Defect	RB,PB	RB,PB
Water	[5%, 15%)	[5%, 15%)
Shape	Circular	Circular
Material	Plastic	Plastic

Task	Ground Truth	CT-GNN
Defect	DE	DE
Water	[5%, 15%)	[5%, 15%)
Shape	Circular	Circular
Material	Lining	Lining

Task	Ground Truth	CT-GNN
Defect	OB,OK	OB,OK
Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular
Material	Iron	Iron

Fig. D.12: Examples of correct classifications with the CT-GNN. Example cases where the CT-GNN correctly classifies all four tasks.



Task	Ground Truth	CT-GNN
Defect	OK	OK,FS
Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular
Material	Plastic	Plastic

Task	Ground Truth	CT-GNN
Defect	BE	BE,OK
Water	[0%, 5%)	[5%, 15%)
Shape	Circular	Circular
Material	Plastic	Plastic

Task	Ground Truth	CT-GNN
Defect	DE,OK	OB,OK
Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular
Material	Lining	Lining

Task	Ground Truth	CT-GNN
Defect	PF,OS	OB,FS,PH
Water	[5%, 15%)	[30%, 100%]
Shape	Conical	Circular
Material	Lining	Concrete

Task	Ground Truth	CT-GNN
Defect	OS	None
Water	[15%, 30%)	[30%, 100%]
Shape	Circular	Conical
Material	Plastic	Lining

Task	Ground Truth	CT-GNN
Defect	OK	None
Water	[5%, 15%)	[0%, 5%)
Shape	Circular	Conical
Material	Plastic	Lining

Fig. D.13: Examples of incorrect classifications with the CT-GNN. Example cases where the CT-GNN incorrectly classifies some or all four tasks. Incorrect classifications are denoted in red.

Table D.14: Water level class examples. Example images of the four considered water level classes.

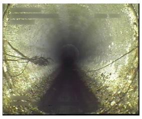
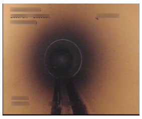
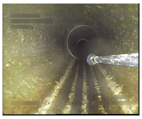
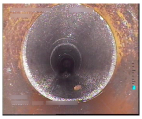
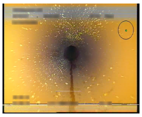
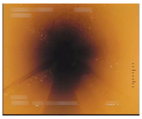
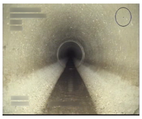
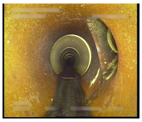
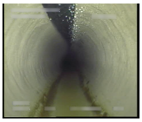
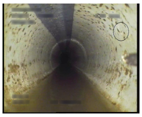
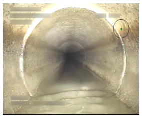
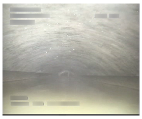
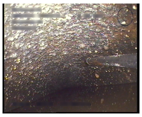
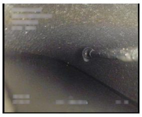
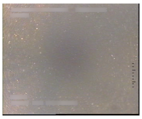
[0%, 5%)					
[5%, 15%)					
[15%, 30%)					
[30%, 100%]					

Table D.15: Pipe shape class examples. Example images of the six considered pipe shape classes.

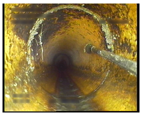
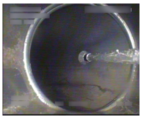
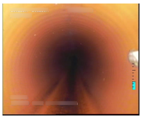
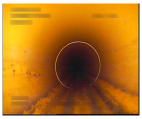
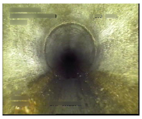
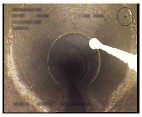
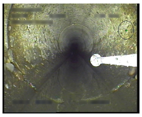
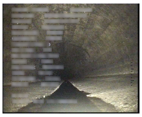
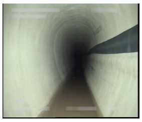
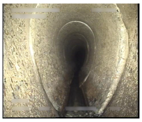
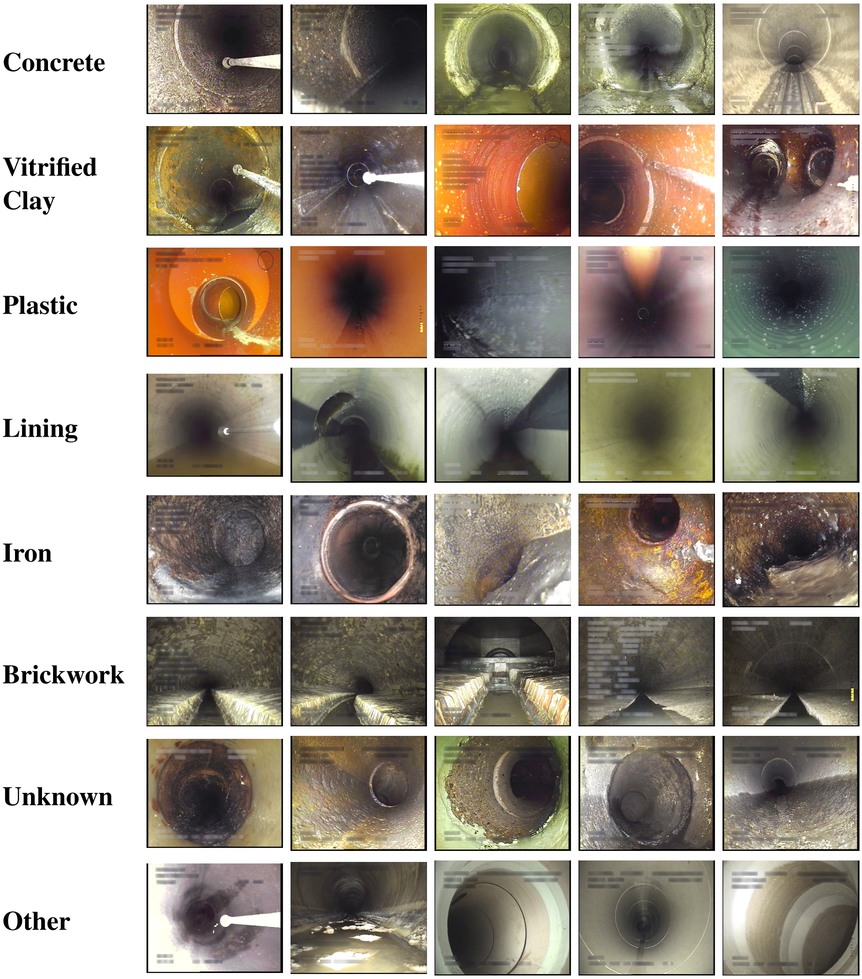
Circular					
Conical					
Egg					
Eye					
Rectangular					
Other					

Table D.16: Pipe material class examples. Example images of the eight considered pipe material classes.



References

[1] J. B. Haurum and T. B. Moeslund, “Sewer-ml: A multi-label sewer defect classification dataset and benchmark,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[2] American Society of Civil Engineers, “2017 infrastructure report card - wastewater,” 2017. [Online]. Available: https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf

[3] European Committee for Standardization, *Investigation and assessment of drain and sewer systems outside buildings – Part 2: Visual inspection coding system*, 1st ed. Dansk Vand og Spildevandsforening (DANVA), 2011.

[4] J. B. Haurum and T. B. Moeslund, “A survey on image-based automation of cctv and sset sewer inspections,” *Automation in Construction*, vol. 111, p. 103061, 2020.

[5] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, “Multi-task learning for dense prediction tasks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[6] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, “Human-level concept learning through probabilistic program induction,” *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.

[7] S.-A. Rebuffi, H. Bilen, and A. Vedaldi, “Learning multiple visual domains with residual adapters,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[8] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[9] S. Vandenhende, S. Georgoulis, and L. Van Gool, “Mti-net: Multi-scale task interaction networks for multi-task learning,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 527–543.

[10] D. Xu, W. Ouyang, X. Wang, and N. Sebe, “Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 675–684.

[11] Z. Zhang, Z. Cui, C. Xu, Y. Yan, N. Sebe, and J. Yang, “Pattern-affinitive propagation across depth, surface normal and semantic segmentation,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4101–4110.

[12] L. Zhou, Z. Cui, C. Xu, Z. Zhang, C. Wang, T. Zhang, and J. Yang, “Pattern-structure diffusion for multi-task learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] P. Guo, C. Deng, L. Xu, X. Huang, and Y. Zhang, “Deep multi-task augmented feature learning via hierarchical graph neural network,” 2020, arxiv: 2002.04813.

[14] Z. Meng, N. Adluru, H. J. Kim, G. Fung, and V. Singh, “Efficient relative attribute learning using graph neural networks,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[15] P. Liu, J. Fu, Y. Dong, X. Qiu, and J. C. Kit Cheung, “Learning multi-task communication with message passing for sequence learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4360–4367, Jul. 2019.

[16] J. Li, P. Zhou, Y. Chen, J. Zhao, S. Roy, Y. Shuicheng, J. Feng, and T. Sim, “Task relation networks,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 932–940.

[17] M. Long, Z. Cao, J. Wang, and P. S. Yu, “Learning multiple tasks with multilinear relationship networks,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[18] Y. Yang and T. M. Hospedales, “Deep multi-task representation learning: A tensor factorisation approach,” in *5th International Conference on Learning Representations, ICLR*, 2017.

[19] D. Alejo, F. Caballero, and L. Merino, “Rgbd-based robot localization in sewer networks,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 4070–4076.

[20] D. Alejo, G. Mier, C. Marques, F. Caballero, L. Merino, and P. Alvito, *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*. Cham: Springer International Publishing, 2020, pp. 275–296.

[21] C. H. Bahnsen, A. S. Johansen, M. P. Philipsen, J. W. Henriksen, K. Nasrollahi, and T. B. Moeslund, “3d sensors for sewer inspection: A quantitative review and analysis,” *Sensors*, vol. 21, no. 7, p. 2553, Apr 2021.

[22] J. B. Haurum, M. M. J. Allahham., M. S. Lynge., K. S. Henriksen, I. A. Nikolov., and T. B. Moeslund., “Sewer defect classification using synthetic point clouds,” in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2021, pp. 891–900.

[23] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, “Generating synthetic point clouds of sewer networks: An initial investigation,” in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2020, pp. 364–373.

[24] S. Iyer, S. K. Sinha, M. K. Pedrick, and B. R. Tittmann, “Evaluation of ultrasonic inspection and imaging systems for concrete pipes,” *Automation in Construction*, vol. 22, pp. 149 – 164, 2012, planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[25] M. S. Khan and R. Patil, “Acoustic characterization of pvc sewer pipes for crack detection using frequency domain analysis,” in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018, pp. 1–5.

[26] ——, “Statistical analysis of acoustic response of pvc pipes for crack detection,” in *SoutheastCon 2018*, 2018, pp. 1–5.

[27] M. Lepot, N. Stanić, and F. H. Clemens, “A technology for sewer pipe inspection (part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification,” *Automation in Construction*, vol. 73, pp. 1 – 11, 2017.

[28] A. D. Tezerjani, M. Mehrandezh, and R. Paranjape, “Defect detection in pipes using a mobile laser-optics technology and digital geometry,” *MATEC Web of Conferences*, vol. 32, p. 06006, 2015.

[29] K. Chen, H. Hu, C. Chen, L. Chen, and C. He, “An intelligent sewer defect detection method based on convolutional neural network,” in *2018 IEEE International Conference on Information and Automation (ICIA)*, Aug 2018, pp. 1301–1306.

[30] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, “Underground sewer pipe condition assessment based on convolutional neural networks,” *Automation in Construction*, vol. 106, p. 102849, 2019.

[31] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr, “Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks,” *Automation in Construction*, vol. 91, pp. 273 – 283, 2018.

[32] D. Li, A. Cong, and S. Guo, “Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification,” *Automation in Construction*, vol. 101, pp. 199 – 208, 2019.

[33] D. Meijer, L. Scholten, F. Clemens, and A. Knobbe, “A defect classification methodology for sewer image sets with convolutional neural networks,” *Automation in Construction*, vol. 104, pp. 281 – 298, 2019.

[34] Q. Xie, D. Li, J. Xu, Z. Yu, and J. Wang, “Automatic detection and classification of sewer defects via hierarchical deep learning,” *IEEE Transactions on Automation Science and Engineering*, pp. 1–12, 2019.

[35] J. C. Cheng and M. Wang, “Automated detection of sewer pipe defects in closed-circuit television images using deep learning techniques,” *Automation in Construction*, vol. 95, pp. 155 – 171, 2018.

[36] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, "Deep learning-based automated detection of sewer defects in cctv videos," *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[37] X. Yin, Y. Chen, A. Bouferguene, H. Zaman, M. Al-Hussein, and L. Kurach, "A deep learning-based framework for an automated defect detection system for sewer pipes," *Automation in Construction*, vol. 109, p. 102967, 2020.

[38] J. Kunzel, T. Werner, P. Eisert, and J. Waschnewski, "Automatic analysis of sewer pipes based on unrolled monocular fisheye images," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 2019–2027.

[39] G. Pan, Y. Zheng, S. Guo, and Y. Lv, "Automatic sewer pipe defect semantic segmentation based on improved u-net," *Automation in Construction*, vol. 119, p. 103383, 2020.

[40] C. Piciarelli, D. Avola, D. Pannone, and G. L. Foresti, "A vision-based system for internal pipeline inspection," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3289–3299, 2019.

[41] M. Wang and J. C. P. Cheng, "A unified convolutional neural network integrated with conditional random field for pipe defect segmentation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, pp. 162–177, 2020.

[42] S. Moradi, T. Zayed, F. Nasiri, and F. Golkhoo, "Automated anomaly detection and localization in sewer inspection videos using proportional data modeling and deep learning-based text recognition," *Journal of Infrastructure Systems*, vol. 26, no. 3, p. 04020018, 2020.

[43] M. Wang, S. S. Kumar, and J. C. Cheng, "Automated sewer pipe defect tracking in cctv videos based on defect detection and metric learning," *Automation in Construction*, vol. 121, p. 103438, 2021.

[44] J. B. Haurum, C. H. Bahnsen, M. Pedersen, and T. B. Moeslund, "Water level estimation in sewer pipes using deep convolutional neural networks," *Water*, vol. 12, no. 12, 2020.

[45] H. Ji, S. Yoo, B.-J. Lee, D. Koo, and J.-H. Kang, "Measurement of wastewater discharge in sewer pipes using image analysis," *Water*, vol. 12, no. 6, p. 1771, Jun 2020.

[46] R. Rothe, R. Timofte, and L. V. Gool, "Dex: Deep expectation of apparent age from a single image," in *IEEE International Conference on Computer Vision Workshops (ICCVW)*, December 2015.

[47] ——, "Deep expectation of real and apparent age from a single image without facial landmarks," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 144–157, 2018.

[48] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, “Detect what you can: Detecting and representing objects using holistic models and body parts,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1979–1986.

[49] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[50] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *Computer Vision – ECCV 2012*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 746–760.

[51] A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[52] S. Ruder, “An overview of multi-task learning in deep neural networks,” 2017, arxiv: 1706.05098.

[53] Y. Zhang and Q. Yang, “A survey on multi-task learning,” 2018, arxiv: 1707.08114.

[54] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich, “GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks,” in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmässan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 794–803.

[55] X. Lin, H.-L. Zhen, Z. Li, Q.-F. Zhang, and S. Kwong, “Pareto multi-task learning,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[56] O. Sener and V. Koltun, “Multi-task learning as multi-objective optimization,” in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds. Curran Associates, Inc., 2018, pp. 525–536.

[57] S. Liu, E. Johns, and A. J. Davison, “End-to-end multi-task learning with attention,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1871–1880.

[58] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[59] L. Liebel and M. Körner, “Auxiliary tasks in multi-task learning,” 2018, arxiv: 1805.06334.

[60] Z. Chen, J. Ngiam, Y. Huang, T. Luong, H. Kretzschmar, Y. Chai, and D. Anguelov, “Just pick a sign: Optimizing deep multitask models with gradient sign dropout,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 2039–2050.

[61] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei, “Dynamic task prioritization for multitask learning,” in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 282–299.

[62] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, “Gradient surgery for multi-task learning,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 5824–5836.

[63] D. Bruggemann, M. Kanakis, S. Georgoulis, and L. V. Gool, “Automated search for resource-efficient branched multi-task networks,” in *Proceedings of the 31st British Machine Vision Conference*, 2020.

[64] P. Guo, C.-Y. Lee, and D. Ulbricht, “Learning to branch for multi-task learning,” in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 3854–3863.

[65] Y. Lu, A. Kumar, S. Zhai, Y. Cheng, T. Javidi, and R. Feris, “Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[66] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, “Which tasks should be learned together in multi-task learning?” in *Proceedings of 37th International Conference on Machine Learning*, 2020.

[67] S. Vandenhende, S. Georgoulis, B. D. Brabandere, and L. V. Gool, “Branched multi-task networks: Deciding what layers to share,” in *Proceedings of the 31st British Machine Vision Conference*, 2020.

[68] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, “Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3200–3209.

[69] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3994–4003.

[70] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4822–4829, Jul. 2019.

[71] D. Bruggemann, M. Kanakis, A. Obukhov, S. Georgoulis, and L. V. Gool, "Exploring relational context for multi-task dense prediction," 2021, arXiv:2104.13874.

[72] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint task-recursive learning for semantic segmentation and depth estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[73] A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas, "Robust learning through cross-task consistency," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 194–11 203.

[74] F. Taherkhani, A. Dabouei, S. Soleymani, J. Dawson, and N. M. Nasrabadi, "Tasks structure regularization in multi-task learning for improving facial attribute prediction," 2021, arXiv:2108.04353.

[75] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, 2020.

[76] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5172–5181.

[77] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, "Attentive single-tasking of multiple tasks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016, pp. 770–778.

[79] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[80] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," *International Conference on Learning Representations*, 2018.

[81] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9260–9269.

Paper E

MSHViT: Multi-Scale Hybrid Vision Transformer and Sinkhorn Tokenizer for Sewer Defect Classification

Joakim Bruslund Haurum, Meysam Madadi, Sergio Escalera, and Thomas B. Moeslund

This paper is submitted to
Machine Vision and Applications.

Abstract

A crucial part of image classification is capturing non-local spatial semantics of image content. In this work, we present MSHViT, a vision transformer based multi-scale extension of the classical CNN backbone, for multi-label sewer defect classification. In order to better model spatial semantics in the images, our approach non-locally aggregate features at different scales through the use of lightweight vision transformer, and produces a smaller set of tokens through a novel Sinkhorn clustering-based tokenizer using distinct cluster centers. We evaluate the proposed MSHViT and Sinkhorn tokenizer on the Sewer-ML multi-label sewer defect classification dataset, showing consistent performance improvements of up to 2.53 percentage points.

1 Introduction

The sewerage infrastructure is one of a few critical infrastructures in modern society. If the infrastructure does not function properly, it can lead to dramatic environmental damage and pose a risk to the public health [1]. Therefore, the sewer pipes require regular inspections in order to determine when a pipe has to be replaced or rehabilitated. However, with more than 1.2 million kilometers of public sewerage infrastructure in just the U.S. [1], this becomes an unimaginable task to perform manually on a regular basis, as each inspection has to be performed by a professional sewer inspector. Therefore, the task of automating the sewer inspection process has been researched for more than three decades, through the development and application of sensors and computer vision algorithms [2–5].

Since its adoption in 2017, the Convolutional Neural Network (CNN) has been the method of choice within the automated sewer inspection domain [2]. A key component of the CNN is the convolutional layers, which efficiently model local spatial semantics within the image. However, for tasks such as multi-label image classification, object detection and object segmentation, it is essential to model non-local spatial semantics [6]. For example, a displaced joint and intruding roots could be simultaneously in an image but in opposite corners. This represents a case where multi-scale non-local spatial semantics are helpful, as knowing the presence of the displaced joint is a strong signal for inferring the presence of the roots.

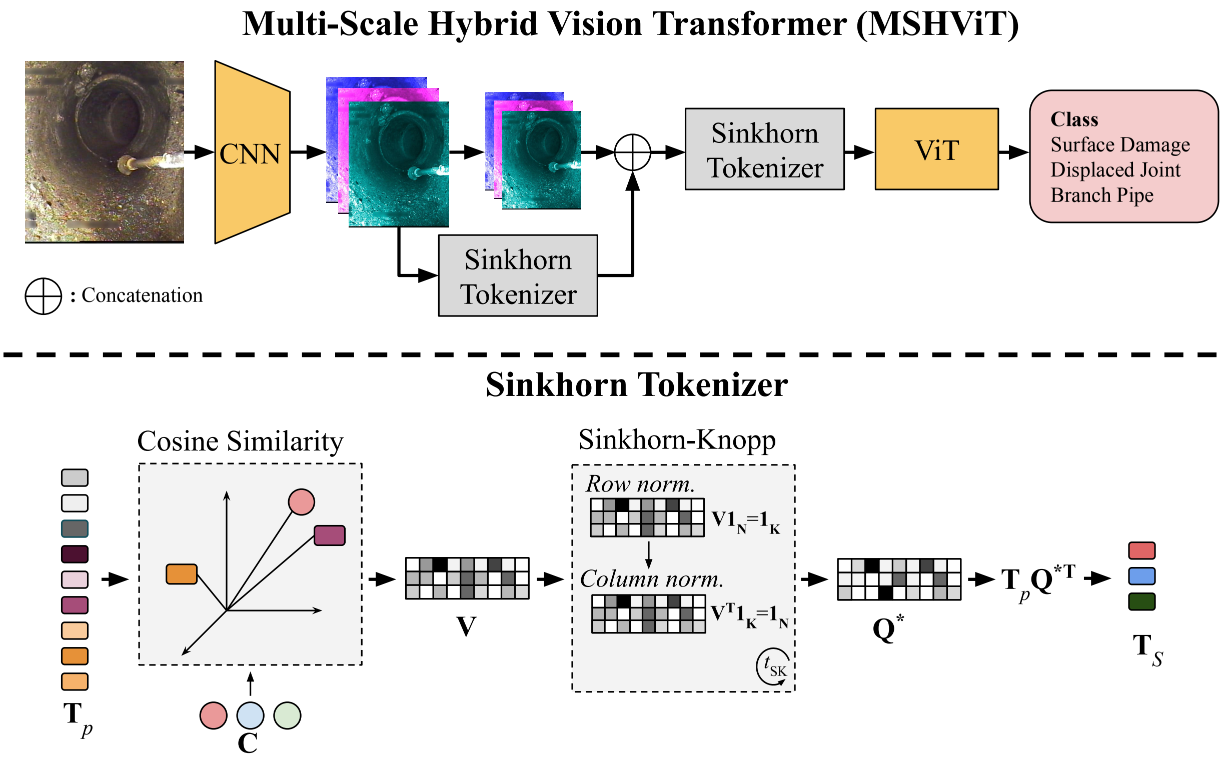


Fig. E.1: System overview. (Top) A CNN backbone returns feature maps from a subset of the internal scales in the CNN. The feature maps from each scale are first tokenized and then processed by a weight-shared ViT. The information from previous scales are propagated forward to the next scale, shown in the figure by forwarding the Sinkhorn tokenizer output to the next scale as per Eq. E.9. (Bottom) The Sinkhorn tokenizer reduces the amount of tokens by first measuring the cosine similarity, $\mathbf{V}$, between all input tokens $\mathbf{T}_p$ and cluster centers $\mathbf{C}$. The Sinkhorn distances [7] are then computed by applying Sinkhorn-Knopp for t_{SK} iterations, resulting in the soft assignment matrix, $\mathbf{Q}^{*\top}$. Using $\mathbf{Q}^{*\top}$ the input features are clustered into the smaller set of tokens, $\mathbf{T}_S$.

Two different approaches have been adopted for vision tasks – either replacing convolutions within the CNN with non-local operations [6, 8–10] or appending CNNs with non-local operations [11–15], denoted Hybrid Vision Transformer (HViT)-like methods in this paper. However, none of these methods explicitly model non-local spatial semantics across scales for image classification, even though it is used as a common approach in object detection and segmentation. We therefore propose the Multi-Scale Hybrid Vision Transformer (MSHViT), where a Vision Transformer (ViT) [13] is appended at different stages of a CNN backbone for non-local aggregation of features and cross-scale propagation of features. We also introduce the Sinkhorn tokenizer, a clustering-based tokenizer to replace the simple patch based tokenizer in ViTs and act as another source of non-local spatial semantics. Furthermore, we demonstrate that the Sinkhorn tokenizer successfully cluster the CNN features, which are expected to have a high amount of redundant information due to successively applying overlapping convolutional filters and pooling layers. We find that introducing these multi-scale and non-local spatial semantics operations leads to a relative improvement compared to using just the CNN backbone.

Our main contributions are as follows:

- We present Multi-Scale Hybrid Vision Transformer (MSHViT), a novel multi-scale extension of the Hybrid Vision Transformer model for capturing non-local spatial semantics across scales.
- We present the Sinkhorn tokenizer, a novel clustering-based tokenizer using Sinkhorn distances, which reduces the amount of tokens and improves metric performance. We visually verify the cross-scale non-local interactions.
- We achieve State-of-the-Art performance on the Sewer-ML multi-label classification sewer dataset, when considering only the defect classification task, outperforming the baseline CNN approaches and other HViT-like approaches.
- We demonstrate the transferability of MSHViT and the Sinkhorn tokenizer across the backbones in the ResNet and TResNet CNN architecture families, and thoroughly investigate the impact of each introduced hyperparameter.

The paper is structured as follows. In Section 2, we review the related works within automated sewer inspections, vision transformers, non-local CNN blocks, and tokenizers. In Section 3, we introduce MSHViT and Sinkhorn tokenizer. In Section 4, we determine the improvement obtained by introducing the MSHViT and Sinkhorn tokenizer and compare to other HViT-like approaches. In Section 5 we conduct an extensive ablation study of the proposed methods, and in Section 6 we qualitatively investigate the clustering assignment made by the Sinkhorn tokenizer. Finally, in Section 7, we conclude the paper.

2 Related Works

In this section we review the literature within the automated sewer inspection domain, as well as recent progress within vision transformers, non-local CNN blocks, and tokenization approaches.

2.1 Automated Sewer Inspections

The automated sewer inspection research field has been active for more than three decades, developing domain specific computer vision algorithms to handle the unique environment that is the sewerage infrastructure [2]. However, the research field has been hindered by the lack of open source code and data, which in combination with differing evaluation protocols, has made it extremely difficult to compare the proposed methods in the literature and caused the field to lag behind the general computer vision domain. This has been rectified within for the classification tasks with the introduction of the public Sewer-ML dataset [16], enabling fair and open comparisons of multi-label classification approaches. Using the Sewer-ML dataset Haurum and Moeslund showed that the sewer defect classification tasks is far from solved. However, the main focus of the field within recent years has been on the defect detection and segmentation tasks [17–24], where no public datasets are available. The field has, however, become more transparent as many have started directly compare different methods on the same datasets, in an effort to offset the lack of public detection and segmentation datasets [20, 22, 24]. Recently, the field has also started investigating other parts of the sewer inspection process [17, 19, 20, 25–29], such as Haurum *et al.* [25] proposing a multi-task classification approach for classifying defects, water level, pipe material, and pipe shape, and Wang *et al.* [17] proposed a framework to accurately determine the severity of defects related to the operation and maintenance of the pipes. The field has also adopted recent trends from the general computer vision field such as self-supervised learning [27], synthetic data generation [30–34], neural architecture search [35], and the usage of the Transformer architecture [20, 36], indicating that the automated sewer inspection field is catching up to the general computer vision domain.

2.2 Vision Transformers

Transformers were originally developed for Natural Language Processing (NLP) [37]. Dosovitskiy *et al.* [13] demonstrated how a pure transformer based architecture, denoted Vision Transformer (ViT), led to competitive performance on several vision classification tasks. The ViT architecture has led to an increased research focus on adapting Transformers for vision tasks [38–48]. A general trend has been introducing components from CNNs into the ViTs, such as limited region of interests and hierarchical representations [40, 43–45] or extending CNNs with transformers in a hybrid approach [12, 13, 15, 38]. However, unlike CNNs the ViT only processes the input image on a single scale due to the original tokenization step and the absence of pooling operations. This problem has been approached in two ways, by either introducing hierarchical representations inspired by classical CNN architecture design [43–46] or multi-scale representations by applying different ViTs sequentially [49] or working on variations of the input in parallel [48, 50]. Our proposed method differs fundamentally from the prior work as we introduce multi scale features by combining CNNs and ViTs, instead of adapting a purely ViT-based model.

2.3 Non-Local CNN Blocks

Combining non-local blocks and operations with classical CNNs have been of great interest as a way of capturing global spatial semantics. The Non-Local Network (NLN) [8] was proposed as an extension of the ResNet architecture family, where non-local aggregation operations were inserted into the last blocks of the architecture. The NLN architecture was extended by Srinivas *et al.* [6] who introduced the Bottleneck Transformer, where Multi-Head Self-Attention was inserted directly into the ResNet bottleneck blocks. Both of these approaches lead to direct improvements on several vision tasks. Appending CNNs with non-local operations have similarly lead to improvements in image classification as shown by Dai *et al.* [14] who investigated how to design Hybrid Vision Transformers (HViTs), *i.e.* CNNs appended with a ViT, as well as in tasks such as object detection with the DETR model [11] and enabling image-caption pair based training [15]. In contrast to the previous application of non-local blocks, we append the CNN at several stages in order to explicitly introduce multi-scale interactions through the proposed MSHViT architecture.

2.4 Tokenizers

An essential part of the transformer architecture is the choice of how to generate the token embedding inputs. In NLP several embedding methods have been utilized through the years in order to represent sentences and words [51, 52]. However, for image data this has not been the case. Dosovitskiy *et al.* [13] proposed simply extracting non-overlapping patches of the input image and linearly map this to an embedding space. This approach has since been iterated upon, by instead extracting overlapping patches [47], learning to select the patch size of the conventional patch tokenizer [53], as well as replacing the initial layer of the Transformer with a convolutional stem similar to those found in CNNs [39]. In parallel different token downsampling approaches have been investigated in order to reduce token redundancy. Goyal *et al.* [54] and Rao *et al.* [55] propose score-based token downsampling methods, where each token is assigned a score based on the incoming attention from other tokens or a predictive subnetwork, respectively. In contrast, this work and the concurrent work by Marin *et al.* [56] proposes a clustering based approach for reducing the amount of tokens. The method by Marin *et al.* utilizes a K-means/medoids based approach, whereas our proposed Sinkhorn tokenizer utilizes Sinkhorn distances [7] in order to softly assign the input tokens to a set of cluster centers. All of the prior approaches [54–56] are focused on pure ViT architectures and inserted in between each encoder block progressively decimating the amount of tokens present. Comparatively, the proposed Sinkhorn tokenizer is applied on HViTs in order to reduce redundancy in the CNN feature-based tokens.

3 Methodology

In this section we first review the Vision Transformer and its hybrid variant originally proposed by Dosovitskiy *et al.* [13]. Then we present our novel clustering-based Sinkhorn tokenizer, designed to reduce the number of redundant tokens in ViTs. Lastly, we present our MSHViT architecture, designed to non-locally combine CNN features at the *i*th scale and progressively combine features across scales, as illustrated in Figure E.1.

3.1 Vision Transformers

The Vision Transformer [13] demonstrated that the original Transformer architecture [37] can be used with little modifications for image classification, and without the image-related inductive biases found in CNNs.

Tokenization

The transformer takes a series of 1D token embeddings as input, and process the series in parallel. In order to convert image data to a series of 1D tokens the input image $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$ is convolved with D different $P \times P$ kernels with a stride of P and flattend to a 1D series of tokens, producing $N = HW/P^2$ linearly embedded tokens $\mathbf{T}_p \in \mathbb{R}^{D\times N}$.

Furthermore, a special class (CLS) token $\mathbf{x}_{\text{CLS}} \in \mathbb{R}^D$ is appended to $\mathbf{T}_p$. The CLS token is randomly initialized and used to generate an image-level feature representation. In order to encode a spatial ordering into the tokens a learnable positional embedding $\mathbf{E}_{\text{pos}} \in \mathbb{R}^{D\times N+1}$ is added, leading to the final token representations:

$$\mathbf{Z}_0 = [\mathbf{x}_{\text{CLS}} \,||\, \mathbf{T}_p] + \mathbf{E}_{\text{pos}}, \tag{E.1}$$

where $||$ denotes concatenation.

ViT Model

The transformer consists of L stacked encoder blocks, each consisting of a token-aggregation step such Multi-Head Self-Attention (MHSA) followed by an inverted bottleneck projecting each token into an intermediate $\mathbb{R}^{D\times r}$ space, where r is an adjustable hyperparamter, followed by a down projection to the D-dimensional feature space. Layer normalization (LN) [57] is applied before both actions, and residual connections around each action. The final feature representation is the CLS token after L blocks and a final layer normalization step, $\mathbf{y} = \text{LN}(\mathbf{Z}_{L,0})$.

Hybrid ViT

Unlike CNNs, ViTs have very little image-specific inductive biases [13]. Therefore, ViTs often require large amount of training data in order to learn relevant relations, which are encoded directly into CNN architectures. However, this lack of inductive biases similarly allows ViTs to learn relations within images, which are not viable with CNNs, such as capturing non-local spatial semantics. The HViT aims at combining these two architectures, by first using a CNN to encode local features, and then compute non-local spatial semantics using a ViT. This is realized by extracting the tokens $\mathbf{T}_p$ from a CNN feature map with a kernel size $P = 1$, typically at the last feature map before the commonly global pooling step. This is in contest to the ViT model where the tokens are extracted directly from the input image $\mathbf{X}$.

3.2 Sinkhorn Tokenizer

The original ViTs generate the token representations of the image through a non-overlapping patch based method [13]. Several methods have been proposed to improve the tokenizer either by reducing the stride of convolutional layer such that the patches overlap [47], or instead use a convolutional stem which aggressively downsamples the spatial dimensions of the input [39]. However, these methods do not consider the redundancy of features encoding similar patches in the image and therefore lead to disproportionately representing these in the generated tokens. While this may be implicitly handled by the attention mechanisms in the ViT, it introduces an unnecessary processing overhead and time needed to learn these relations.

To deal with the redundant features we introduce a clustering-based tokenizer using Sinkhorn distances [7], inspired by clustering-based self-supervised learning [58, 59]. The approach builds upon the original patch tokenizer with $P = 1$. The N patch tokens $\mathbf{T}_p$ are compared to K cluster centers $\mathbf{C} \in \mathbb{R}^{D \times K}$ which are initialized from a D-dimensional Normal distribution with zero-mean and unit variance. We assume both $\mathbf{T}_p$ and $\mathbf{C}$ are ℓ_2 normalized and measure similarity using the cosine similarity $\mathbf{V} = \mathbf{C}^\top \mathbf{T}_p \in \mathbb{R}^{K \times N}$. Based on the similarity scores $\mathbf{V}$ we compute the soft assignment matrix $\mathbf{Q} \in \mathbb{R}_+^{K \times N}$, which belongs to the set of valid assignment matrices $\mathcal{Q}$, such that the similarity between the cluster centers and features is maximized:

$$\max_{\mathbf{Q} \in \mathcal{Q}} \operatorname{Tr}(\mathbf{Q}^\top \mathbf{V}) + \varepsilon H(\mathbf{Q}), \tag{E.2}$$

where H is the matrix entropy function and ε controls the weighting of the entropy loss and thereby the smoothness of the assignment scores.

Similar to [58, 59] we constrain $\mathbf{Q}$ to be in the transportation polytope under an equipartition constraint of the input and cluster centers *i.e.* the features should on average be uniformly assigned to the cluster centers. However, instead of applying the constraint on the full dataset [58] or mini-batches [59], we apply the constraint on the N features from a single input, see Eq. E.3. We apply the constraint on the N features such that there are no cross-information between input images, enabling single image evaluation.

$$\begin{aligned}\mathcal{Q} = &\{\mathbf{Q} \in \mathbb{R}_{+}^{K\times N} | \\ &\mathbf{Q}\mathbf{1}_N = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{N}\mathbf{1}_N\},\end{aligned} \tag{E.3}$$

where $\mathbf{1}_K$ and $\mathbf{1}_N$ are K and N-dimensional vectors filled with ones, respectively.

The solution to Eq. E.2 can then be formulated as follows:

$$\mathbf{Q}^* = \operatorname{diag}(\mathbf{u}) \exp\left(\frac{\mathbf{V}}{\varepsilon}\right) \operatorname{diag}(\mathbf{v}) \in \mathbb{R}_{+}^{K\times N}, \tag{E.4}$$

where the renormalization vectors $\mathbf{u}$ and $\mathbf{v}$ are computed using the iterative Sinkhorn-Knopp algorithm [7] through t_{SK} iterations.

Using the soft assignments between input features $\mathbf{T}_p$ and cluster centers $\mathbf{C}$ stored in $\mathbf{Q}^*$ we transform the input features into K new tokens:

$$\mathbf{T}_S = \mathbf{T}_p \mathbf{Q}^{*\top} \in \mathbb{R}^{D\times K} \tag{E.5}$$

3.3 Multi-Scale Hybrid Vision Transformers

Based on prior work on combining non-local operations with classical CNNs, such as HViTs, we propose the Multi-Scale Hybrid Vision Transformer. Whereas the original HViT simply extends the backbone CNN with a ViT, we propose applying ViTs at different scales of the backbone CNN. Furthermore, we also introduce cross-scale connections between the ViTs in order to encode non-local spatial semantics in the image at different scales, see Figure E.1.

CNNs such as ResNets [60] and Inception networks [61, 62] have a set of natural scales within them due to the periodic pooling operations. The representative feature map of each scale is defined to be the last feature map before each pooling operation and denoted $\mathbf{X}^i$ for the ith scale. At every scale each feature in $\mathbf{X}^i$ is linearly embedded into a common D-dimensional space as tokens $\mathbf{T}_p^i$. These tokens are processed using a tokenization function ψ^i, representing either the Sinkhorn tokenizer (Eq. E.5) or an identity function for the standard patch tokenizer, with the final scale tokens denoted $\mathbf{T}^i$. The tokens can then be processed by a scale-specific ViT of depth L, denoted as ϕ^i, producing the scale features:

$$\mathbf{Z}_L^i = \phi^i(\mathbf{T}^i) \tag{E.6}$$

Cross-Scale Connections

In order to share information between different scales, we introduce cross-scale connections. For scale $i > 1$ all or a subset of the previous scale features are included, denoted $\mathbf{S}^i$, in addition to the ith scale features $\mathbf{T}_p^i$, see Eq. E.7.

$$\mathbf{T}^i = \psi^i(\mathbf{T}_p^i \,||\, \mathbf{S}^i) \tag{E.7}$$

This cross-scale connection can occur using features from three different stages: the CNN features $\mathbf{T}_p$, see Eq. E.8, the initial tokens $\mathbf{T}$, see Eq. E.9, or the final token embeddings $\mathbf{Z}_L$, see Eq. E.10. j denotes the initial scale which we consider for scale i. For example, if $j = 1$ all features from scale 1 to scale $i-1$ are aggregated, while if $j = i-1$ only the features from scale $i-1$ are aggregated.

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{T}_p^j \tag{E.8}$$

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{T}^j \tag{E.9}$$

$$\mathbf{S}^i = \|_j^{i-1} \mathbf{Z}_L^j \tag{E.10}$$

Lastly, the overall image representation is defined to be $\mathbf{y} = \mathrm{LN}(\mathbf{Z}_{L,0}^I)$, where I denotes the last scale of the backbone.

4 Experimental Results

In this section we investigate the performance of the MSHViT architecture and Sinkhorn tokenizer on the Sewer-ML dataset, a multi-label sewer defect classification dataset [16]. Sewer-ML is the world's only public multi-label sewer defect dataset, consisting of 1.3 million images, 17 defect classes, and the implicit normal class. The dataset is split into three distinct training, validation, and testing splits, each containing 1 million, 130k and 130k images, respectively. We refer to the supplementary material of Haurum and Moeslund [16] for example images. Defect predictions are evaluated using the F2-score weighted by the classes *class importance weight* (CIW), $\mathrm{F2}_{\mathrm{CIW}}$, which indicates the economic importance of the classes, and the normal pipes are evaluated by the F1-score, $\mathrm{F2}_{\mathrm{CIW}}$ [16].

4.1 Training Procedure

We follow the training procedure of Haurum *et al.* [25] with the addition of using the Exponential Moving Average (EMA) technique on the model weights, see Table E.1. We utilize the Fourier Network (FNet) based attention mechanism [63] in the HViT as an efficient alternative to the conventional MHSA based attention mechanism.

We define the ResNet architecture to have five natural scales: the convolutional stem followed by four ResNet blocks, numbered from 1 to 5. These stages are chosen as they act on feature maps with different spatial dimensions.

4.2 Hyperparameter Search

The hyperparameter search for the MSHViT and Sinkhorn tokenizer are conducted in a sequential manner in order to reduce the search space due to the amount of hyperparameters and the investigated value ranges. The investigated hyperparameter values as well as the initial and final values are shown in Table E.2. The initial Sinkhorn

Table E.1: Detailed training procedures. We follow the training procedures of Haurum *et al.* [25] with the addition of utilizing model EMA.

Variable	**Value**
Image Size	224
Epochs	40
Batch Size	258
Learning Rate (LR)	0.1
Weight Decay	0.0001
LR Scheduler	Step @ 20, 30 epochs
LR Decay Factor	0.01
Optimizer	SGD w/ momentum
Loss function	Binary Cross-Entropy
Class Weighting	Effective samples [64] $\beta = 0.9999$
Model EMA	0.9997
Augmentations	Horizontal flip ($p = 0.5$) Color Jitter (± 0.1)

Tokenizer values were set as in Caron *et al.* [59], except for the number of clusters K, where we chose 64 centers as the initial value to ensure a large average assignment probability per cluster in each scale. For the MSHViT architecture we initialized the model with the last two layers, where higher-order features are available. The hyperparameters of the ViTs were chosen such that only a moderate parameter increase was introduced. After each step in the sequential search we used the configuration which performed the best for the next step. The steps of the sequential search were ordered such that the Sinkhorn Tokenizer cluster and MSHViT cross-scale hyperparameters were determined, and lastly the structure of the ViTs. The order of the search was as follows:

1. Search over the entropic regularization ε in the Sinkhorn tokenizer.
2. Search over the number of iterations t_{SK} in the Sinkhorn tokenizer.
3. Search over the number of clusters K in the Sinkhorn tokenizer.
4. Search over which scales to be used and selection of j in the MSHViT extension.
5. Search over the multi-scale method, $\mathbf{S}$.
6. Search over token dimensionality D.
7. Search over the MLP ratio r.
8. Search over vision transformer depth L.

We find that the initial hyperparameters perform well, with only the entropic regularization and number of iterations in the Sinkhorn-Knopp algorithm being adapted.

Table E.2: Hyperparameters. Overview of all searched hyperparamters, with the investigated values as well as the initial and final values.

HP	Range	Initial	Final
ε	[0.05, 0.25, 0.5, 0.75, 1.00, 1.25]	0.05	0.25
t_{SK}	[1, 3, 5, 7, 9]	3	5
K	[32, 64, 128, 64/32, 128/64]	64	64
Scales	[{2,3,4,5}, {3,4,5}, {4,5}, {5}]	{4,5}	{4,5}
$\mathbf{S}$	$[\mathbf{T}_p, \mathbf{T}, \mathbf{Z}_L]$	$\mathbf{T}$	$\mathbf{T}$
j	$[i-1, \min(\text{Scales})]$	$i-1$	$i-1$
D	[512, 1024, 2048]	512	512
r	[1, 2, 3, 4]	4	4
L	[1, 2, 3]	2	2

4.3 Comparative Models

We investigate the performance increase incurred when applying MSHViT to the ResNet-{18, 34, 50, 101}, a commonly used backbone architecture in the image classification literature [6, 12, 65], as well as TResNet backbone [66], an adaption of the ResNet backbone using concepts such as anti-aliased downsampling and Squeeze and Excitation (SE) [67] layers. Furthermore, we compare performance against the HViT-like models BoTNet-50-S1 [6] and CoAtNet-{0,1} [14], as well as the original HViT structure [13]. BotNet and CoAtNet were trained with the model structure described in the original papers, while the HViT model uses the same ViT parameters described in Table E.2 with the exception of the attention mechanism where we use the classical MHSA based token mixing. We compare using both the conventional patch based tokenizer and the proposed Sinkhorn tokenizer. Lastly, we compare to the previously published results on Sewer-ML [16, 25]. We run all experiments within the same codebase, using the torchvision [68], Pytorch Lightning [69] and timm [70] libraries. All models were trained using a single Nvidia V100 GPU except for the CoAtNet models which required two V100 GPUs due to a higher VRAM consumption.

4.4 Results

We find that introducing the MSHViT and Sinkhorn Tokenizer leads to a noticeable improvement on all tested backbones, see Table E.3. On the F2_{CIW} metric we observe an increase of 0.7 and 2.5 percentage points, with the largest increase observed on the ResNet-50, where the performance is improved by 2.4-2.5 percentage points on both the validation and testing splits. On the $\text{F1}_{\text{Normal}}$ we observe a more moderate increase of up to 0.24 percentage points. However, we observe a generally higher baseline performance compared to previous methods. This is a comparable performance to the previous state-of-the-art on Sewer-ML, the multi-task classification method CT-GAT, while only using defect labels.

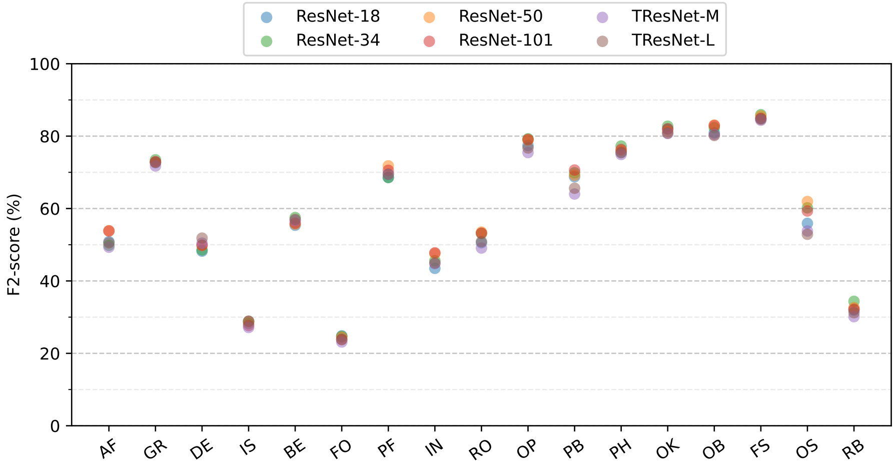
Interestingly, we observe that the ResNet-34 backbone performance surprisingly well for both the baseline and MSHViT extension. Not only does the ResNet-34 baseline achieve the best performance out of the ResNet networks, it also either outperforms or matches the ResNet-101 backbone when applying the MSHViT extension. For the TResNet architectures we observe that the improvement gained by adding MSHViT extension is smaller than that for the ResNet backbones. This is most likely due to the SE layers in the TResNet model, which means the TResNet already includes some attention based mechanisms. However, it is clear that the MSHViT extension is still beneficial.

When comparing to other HViT-like models we see that the MSHViT extension outperforms the original HViT structure, as well as all models where the transformer structure is incorporated directly into the backbone. It should be note that on the validation split the BotNet-50-S1 model nearly matches the ResNet-50-MSHViT's F2_{CIW} and achieves state-of-the-art $\text{F1}_{\text{Normal}}$ performance. However, on the test split the F2_{CIW} performance is significantly lowered compared to the ResNet-50-MSHViT, indicating the model does not generalize as well as the ResNet-50-MSHViT model.

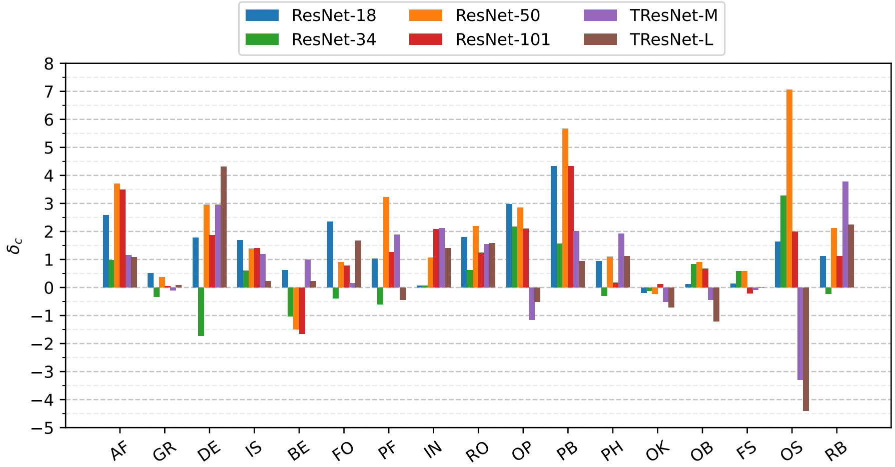
From these results we can conclude that the proposed MSHViT extension lead to improvements without tuning the hyperparameters for the backbone. We hypothesize that if hyperparameters were tuned for each backbone, the performance gain would further increase.

Table E.3: Results on Sewer-ML. Comparison using the investigated CNN backbones. We compare each backbone with and without the MSHViT and Sinkhorn tokenizer extension (denoted MSHViT) using the F2_{CIW} and $\text{F1}_{\text{Normal}}$ metrics [16]. Best performance per column is denoted in **bold**. We also include the previous published results on Sewer-ML [16, 25], and HViT-like models [6, 13, 14]. * denotes that the method was trained in a multi-task classification framework.

Model		Validation Split		Test Split	
Model	**MSHViT**	**F2_{CIW}**	**$\text{F1}_{\text{Normal}}$**	**F2_{CIW}**	**$\text{F1}_{\text{Normal}}$**
Benchmark [16]	-	55.36	91.32	55.11	90.94
CT-GAT* [25]	-	**61.70**	91.94	**60.57**	91.61
ResNet-50-HViT-Patch [13]	-	59.87	92.41	57.58	91.99
ResNet-50-HViT-Sinkhorn [13]	-	60.42	92.41	58.74	92.07
BotNet-50-S1 [6]	-	61.62	**92.92**	59.69	**92.49**
CoAtNet-0 [14]	-	57.82	92.28	56.53	91.94
CoAtNet-1 [14]	-	59.37	92.50	57.42	91.11
ResNet-18 [60]	✗	58.60	92.34	56.62	91.88
	✓	59.87	92.42	58.18	92.12
ResNet-34 [60]	✗	60.98	92.72	59.18	92.30
	✓	61.65	92.76	59.91	92.30
ResNet-50 [60]	✗	59.28	92.44	57.58	92.03
	✓	61.68	92.44	60.11	92.11
ResNet-101 [60]	✗	60.06	92.48	58.01	92.13
	✓	61.25	92.50	59.93	92.19
TResNet-M [66]	✗	58.04	92.22	56.08	91.90
	✓	58.68	92.25	56.93	91.84
TResNet-L [66]	✗	59.17	92.36	56.97	92.00
	✓	59.19	92.27	57.16	91.87



(a) Per-class performance for all MSHViT models.



(b) Per-class difference when comparing the MSHViT and baseline models.

Fig. E.2: Per-Class F2-scores analysis. We present the per-class F2-scores on the validation split for all MSHViT-based models as well as the difference between the MSHViT variants and the baseline models, δ_c. The classes are sorted in ascending order by their class-importance weight [16]. Class names and abbreviations are described in the original Sewer-ML paper [16].

4.5 Per-Class Analysis

In order to better understand how the compared models work, we investigate how the baseline and MSHViT extended models differ in their class predictions on the validation split. In Figure E.2a we present the per-class F2-scores for all MSHViT models, and in Figure E.2b we determine the difference in per-class F2-scores when comparing the MSHViT variants with the baseline models, see Eq. E.11.

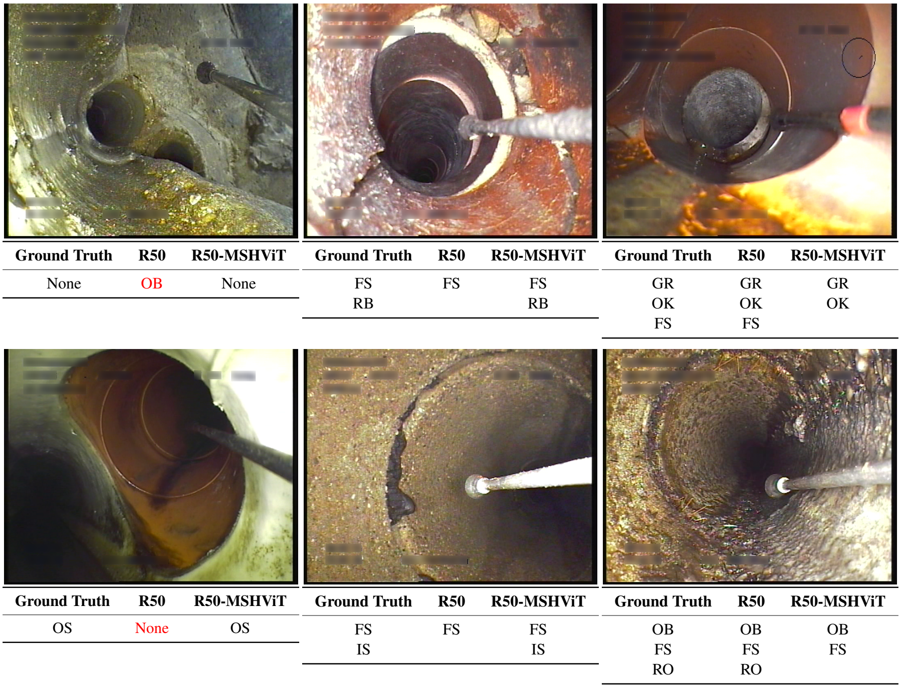
$$\delta_c = c_{\text{MSHViT}} - c_{\text{Baseline}}, \tag{E.11}$$

where δ_s^c is the difference in F2-scores for class c, and c_{MSHViT} and c_{Baseline} are F2-scores for class c for the MSHViT and Baseline models, respectively.

When analyzing the absolute per-class performance in Figure E.2a, we see that the ResNet-34, ResNet-50, and ResNet-101 all perform similarly well on nearly all classes, with the ResNet-34 and ResNet-50 achieving noticeable performances in the highest weighted classes, whereas the TResNet models and ResNet-18 have a noticeably lower score on several classes. In Figure E.2b observe that when using MSHViT together with the ResNet backbones performance increases on nearly all classes, except for consistent decreases on the attached deposits (BE) class and on the connection with construction changes (OK) class. For the ResNet-34 backbone we also observe a significant decrease in performance on the deformation (DE) class. However, there is a noticeable increase in performance on both the lateral reinstatement cuts (OS) and cracks, breaks, and collapses (RB), the two highest weighted classes, across all ResNet backbones. On the contrary we see that the TResNet backbones behaves very poorly on the OS class, which drags down the overall score, even though it performs well on nearly all other classes.

4.6 Qualitative Examples

In addition to quantiative per-class comparison, we also look into specific cases where the predictions of the compared models differ. Focusing on the ResNet-50 backbone we compare cases where the MSHViT extensions matches all classes correctly while the baseline misclassifies some or all classes and vice versa, see Figure E.3. Four examples are shown where the MSHViT model correctly predicts all classes. In the top left image, the MSHVIT correctly predicts the pipe to be normal, whereas the baseline predicts surface damage (OB). This is most likely due to the missing top half of the pipe, as the image is taken from within the sewer well. In the top middle and bottom left cases the baseline misses the cracks, breaks, and collapses (RB) and lateral reinstatement cuts (OS) classes, the two highest weighted classes by CIW. Missing these classes could lead to significant economic repercussions. The RB class is most likely missed due to its visual similarity to the displaced joint (FS) deeper in the pipe, whereas the OS is similarly missed as the baseline misses the fact that a lining has been inserted and the low severity of the class. In the bottom middle example, the baseline simply misses the intruding sealing material (IS) class, instead only classifying the displaced joint (FS). In the top right and bottom right, the MSHViT variant misses the displaced joint (FS) and roots (RO), respectively. It is not clear why the MSHViT missed the displaced joint, however, we hypothesize it might be due to the co-occurring connection with construction changes (OK) class, where the material of the pipe changes. For the bottom right case, the MSHVIT misses the small fine roots in the joint, most likely due to focusing on the much more prevalent displaced joint (FS) and surface damage (OB).



Ground Truth	R50	R50-MSHViT
None	OB	None

Ground Truth	R50	R50-MSHViT
FS	FS	FS
RB		RB

Ground Truth	R50	R50-MSHViT
GR	GR	GR
OK	OK	OK
FS	FS	

Ground Truth	R50	R50-MSHViT
OS	None	OS

Ground Truth	R50	R50-MSHViT
FS	FS	FS
IS		IS

Ground Truth	R50	R50-MSHViT
OB	OB	OB
FS	FS	FS
RO	RO	

Fig. E.3: Examples of classifications with MSHViT. Example cases where the MSHViT extensions correctly classifies all classes as well as misclassifies some classes. The class codes are described in the original Sewer-ML paper [16]. Incorrect predictions are shown in red.

4.7 Efficiency Analysis

In order to verify that the increased metric performance is not simply due to an increase in learnable parameters, we compare the validation $F2_{\text{CIW}}$ against the number of trainable parameters in the models as well as the throughput measured in images processed per second (img/s) during both training and inference, as recommended by Dehghani *et al.* [71]. The throughput performance is computed over 200 batches of 256 images with an initial 10 warmup batches, and averaged over five separate runs. As the method from Haurum and Moeslund [16] is a two-stage approach and the method from Haurum *et al.* [25] is designed for the multi-task classification task, we do not include these in the throughput comparison. The results are shown in Figure E.4. From these results it is clear that the increased performance obtained with the MSHViT extension is not only due to the increase amount of parameters, as the extended models consistently outperforms baseline variants with a higher number of parameters. When looking at the throughput of the models, we see that the MSHViT does lead to a slower processing speed, however, for the larger models such as ResNet-50 and ResNet-101 this slowdown is marginal at best.

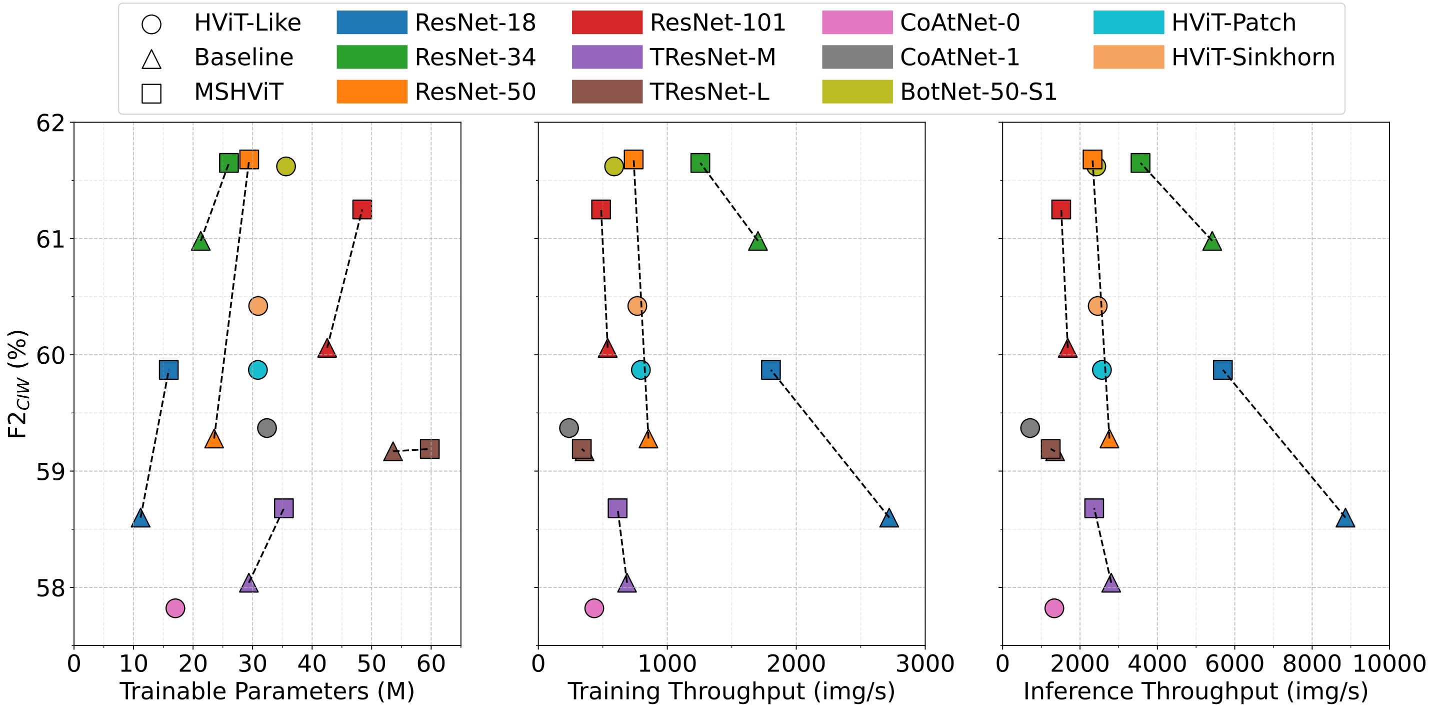


Fig. E.4: Comparison of metric performance and efficiency. We compare the performance of the models in Table E.3 against the parameter count of each model as well as the models throughput in image per second (img/s) during training and inference. MSHViT variants are linked to their baseline variant by a dotted line.

Table E.4: Effect of ε. Comparison of different entropic regularization values in the Sinkhorn tokenizer.

ε	$\mathbf{F2_{CIW}}$	$\mathbf{F1_{Normal}}$
0.05	60.80	**92.56**
0.25	**61.68**	92.44
0.50	61.33	92.47
0.75	60.85	92.35
1.00	60.86	92.51
1.25	60.46	92.36

5 Ablation Studies

We conduct a series of ablations studies in order to determine the sensitivity to the hyperparameter settings in the Sinkhorn tokenizer and MSHViT architecture. All tests are conducted on the Sewer-ML validation set using a ResNet-50 backbone, with the hyperparameter values stated in Table E.1-E.2 unless otherwise stated.

5.1 Sinkhorn-Knopp Hyperparameters

At the heart of the Sinkhorn tokenizer is the iterative Sinkhorn-Knopp algorithm, which is controlled by two hyperparameters: t_{SK} and ε. We investigate these hyperparameters' effect on the metric performance one at a time.

First, we investigate the strength of the entropic regularization term in Eq. E.2 comparing values of $\varepsilon = \{0.05, 0.25, 0.50, 0.75, 1.00, 1.25\}$, see Table E.4. We observe that the highest $\mathrm{F2_{CIW}}$ and $\mathrm{F1_{Normal}}$ are achieved using $\varepsilon = 0.25$, a slightly higher entropic regularization term than what has previously been used in the self-supervised training domain [59]. In general, we see that a too high or low entropic regularization negatively affects the $\mathrm{F2_{CIW}}$ performance.

Secondly, we investigate the effect of the number of iterations conducted t_{SK}. We compare the performance when setting $t_{\mathrm{SK}} = \{1, 3, 5, 7, 9\}$, see Table E.5, as well as the effect on efficiency by measuring training and inference img/s, see Figure E.5. We observe that peak performance on both $\mathrm{F2_{CIW}}$ and $\mathrm{F1_{Normal}}$ is achieved when t_{SK} is set to 5, while too few or too many iterations led to degradation in performance. We also observe a monotonic decrease in throughput when t_{SK} is increased, as expected. When compared to the conventional patch tokenizer we observe that the training throughput and the inference throughput of the Sinkhorn tokenizer beats that of the patch tokenizer at all settings of t_{SK}.

5.2 Number of Cluster Centers K

A key part of the Sinkhorn tokenizer is the number of clusters K. We investigate the effect of setting $K = \{32, 64, 128, 64/64, 128/64\}$, where x/y denotes x clusters for

Table E.5: Effect of t_{SK}. Comparison of number of iterations in the Sinkhorn tokenizer.

t_{SK}	F2_{CIW}	$\text{F1}_{\text{Normal}}$
1	61.16	**92.58**
3	61.24	92.47
5	**61.68**	92.44
7	61.13	92.50
9	61.45	92.47

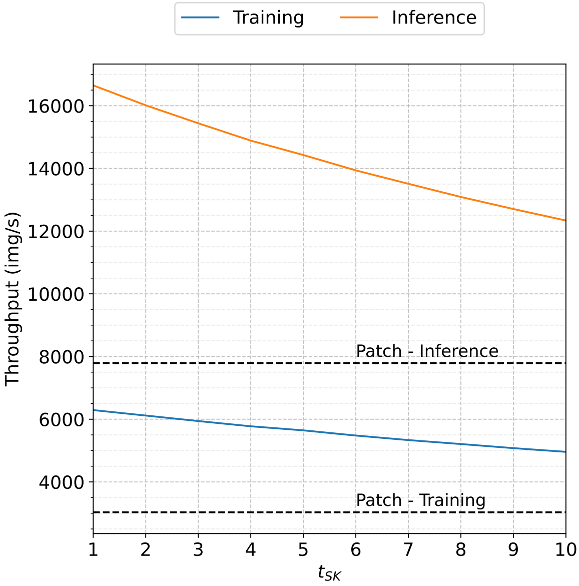


Fig. E.5: Effect of t_{SK} on throughput. Comparison of the training and inference throughput at different amount of iterations in the Sinkhorn tokenizer, t_{SK}. Training and inference throughput is also shown for the conventional patch tokenizer. Throughput is measured only for the MSHViT extension, as the backbone processing time is simply an offset.

the 4th scale and y clusters for the 5th scale, see Table E.6. We find that increasing or decreasing the number of cluster centers slightly reduced the classification performance, whereas having more clusters for earlier scales dramatically decreased performance. This is hypothesized to be due to the earlier clusters capturing similar semantics, as the larger amount of cluster centers allow a less aggressive clustering process.

5.3 Tokenizer Efficiency at Different Image Resolutions

A key benefit of the Sinkhorn Tokenizer is the constant efficiency when the image resolution is increased. To demonstrate this we compare the training and inference

Table E.6: Effect of number of cluster centers. Comparison of metric performance when varying the number of cluster centers K in the Sinkhorn tokenizer.

K	$\mathbf{F2_{CIW}}$	$\mathbf{F1_{Normal}}$
32	61.33	92.47
64	**61.68**	92.44
128	61.33	92.34
64/32	60.56	92.46
128/64	60.73	**92.54**

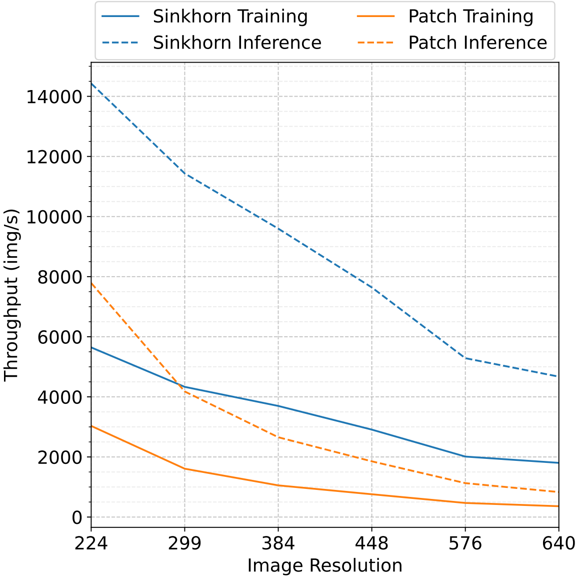


Fig. E.6: Effect of image resolution on throughput. We compare the training and inference throughput for the Sinkhorn and patch tokenizers across commonly used image resolutions. The Sinkhorn tokenizer consistently achieves a higher throughput than the conventional patch tokenizer. Throughput is measured only for the MSHViT extension, as the backbone processing time is simply an offset

throughput of the MSHViT model (excluding the backbone, which would simply be an offset) at different image resolutions, when using the conventional patch tokenizer and the proposed Sinkhorn tokenizer, see Figure E.6. From this it is clear that the throughput of the Sinkhorn tokenizer better handles the changes in image resolutions, whereas the throughput of the conventional patch tokenizer suffers greatly when the resolution is increased.

Table E.7: Effect of ℓ_2 normalization. Comparison of performance when ℓ_2 normalizing the cluster centers $\mathbf{C}$ and input features $\mathbf{T}_p$, before computing the similarity scores $\mathbf{V}$.

ℓ_2 **normalized**	F2_{CIW}	$\text{F1}_{\text{Normal}}$
✗	60.40	92.40
✓	**61.68**	**92.44**

Table E.8: Effect of sharing tokenizer. Comparison of metric performance when sharing tokenizer cluster centers.

Shared Tokenizer	F2_{CIW}	$\text{F1}_{\text{Normal}}$
✗	**61.68**	92.44
✓	61.22	**92.53**

5.4 Effect of ℓ_2 Normalization

Within the Sinkhorn-Knopp algorithm is the calculation of the cosine similarities between cluster centers and input features, $\mathbf{V}$, This step requries an ℓ_2 normalization of all cluster centers and input features in order to yield output values between 0 and 1. We investigate the effect of skipping this normalization step, see Table E.7. We see that the metric performance clearly drops when the features are not normalized onto the unit D-sphere. We can therefore conclude the normalization step is crucial for the Sinkhorn tokenizer.

5.5 Effect of Shared Sinkhorn Tokenizer

Inspired by the Perceiver papers [72, 73] we investigate the performance when sharing the tokenizer cluster centers and linear projection weights, see Table E.8. We find that when sharing the tokenizer parameters, the performance decreases by nearly a half percentage point. This is expected as the same cluster centers have to meaningfully represent CNN features from all considered scales, even though the CNN features are hierarchical in nature.

5.6 Comparison of Attention Mechanisms and Tokenizers

We investigate whether the Sinkhorn tokenizer leads to improvements compared to the standard non-overlapping tokenizer from Dosovitskiy *et al.* [13], as well as the effect of attention mechanism, see Table E.9. The patch based tokenizer uses a kernel size and stride of $P = 1$ for both scales. We observe that the Sinkhorn tokenizer outperforms the conventional patch tokenizer on all attention mechanisms, and that the inverted bottleneck yields little benefit in all cases but the Sinkhorn tokenizer combined with FNet. This shows a clear benefit from the clustering-based Sinkhorn tokenizer.

Table E.9: Effect of tokenizer and attention mechanism. Comparison of metric performance when using the standard non-overlapping patch tokenizer and the Sinkhorn tokenizer. #P indicates the amount of trainable parameters in the MSHViT head in millions.

Attention	Tokenizer	#P	$F2_{CIW}$	$F1_{Normal}$
Fourier	Patch	1.72	59.61	92.46
	Sinkhorn		61.03	92.41
MHSA	Patch	3.82	58.95	92.24
	Sinkhorn		61.09	**92.49**
FNet	Patch	5.92	59.46	92.37
	Sinkhorn		**61.68**	92.44
Transformer	Patch	8.02	59.20	92.38
	Sinkhorn		61.11	92.41

Table E.10: Effect of using different scales. Comparison of metric performance when using different scales and different cross-scale sharing range j.

Scales	j	$F2_{CIW}$	$F1_{Normal}$
2, 3, 4, 5	$i-1$	61.36	92.49
3, 4, 5	$i-1$	60.92	92.44
4, 5	$i-1$	**61.68**	92.44
2, 3, 4, 5	2	60.45	92.49
3, 4, 5	3	60.86	92.37
5	-	61.03	**92.52**

5.7 Effect of Multi-Scale Approach

In order to determine the effect of the multi-scale approach, we compare the performance when using different scales and the range of the cross scale connections j. Specifically, we compare using subsets of the scales 2-5 of the ResNet architecture *i.e.* all but the convolutional stem scale, as well as cross scale connections with $j = i-1$ where only the previous scale is relevant, or j set equal to the initial scale. The comparison is listed in Table E.10, where it is clear that a multi-scale approach outperforms the classic single-scale HViT architecture, and that using too many scales diminish the performance.

5.8 Comparison of Cross-Scale Connections

A key part of the MSHViT architecture is the multi-scale connections which enable information sharing across scales. Three variations are presented in Eq E.8-E.10, and compared in Table E.11. We also compare against a scenario with no cross-scale

Table E.11: Comparison of cross-scale mechanisms Comparison of metric performance when using a late stage scale fusion step or cross-scale mechanism **S** (Eq. E.7) using either CNN (Eq E.8), Sinkhorn tokenizer (Eq E.9), or ViT (Eq E.10) features.

S	**Shared ViT**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**
-	✗	59.88	92.31
-	✓	60.06	92.40
$\mathbf{T}_p$	-	60.25	92.38
$\mathbf{T}$	-	**61.68**	92.44
$\mathbf{Z}_L$	✗	60.75	**92.49**
$\mathbf{Z}_L$	✓	61.37	92.48

information sharing between the ViTs, instead using a late stage scale-fusion step. The late stage fusion step combines the CLS tokens from each scale together with a learnable cross-scale CLS token, using a MHSA operation with 8 heads. We find that all cross-scale connections outperform the late stage scale-fusion variation and that using the ViT or linearly embedded CNN features led to a decrease in metric performance. Instead the best performance is achieved by sharing the clustered tokens from the Sinkhorn tokenizer across scales, indicating that the clustering process is crucial for performance. We also compare sharing weights for the ViTs when applicable, and find that sharing ViT weights results in a clear performance benefit, unlike when sharing weights and cluster centers in the tokenizer (See Section 5.5).

5.9 Effect of ViT Hyperparameters

Lastly, we investigate the effect of varying the hyperparameters of the ViT. Specifically, we investigate the effect of the token dimensionality, D, the MLP ratio, r, in the inverted bottleneck, and the depth of the ViT, L. The effect on the metrics are reported in Table E.12-E.14, as well as the number of trainable parameters in the MSHViT extension, #P. From these results we observe a clear decrease in metric performance when increasing the token dimensionality D, as well as when the ViT is too shallow or deep. For the MLP ratio we observe that best performance is achieved when $r = 4$, with performance in general decreasing when lowering r as the inverted bottleneck becomes narrower.

Table E.12: Effect of token dimensionality D**.** We see that increasing the token dimensionality leads to poorer performance.

D	**#P**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**
512	5.92	**61.68**	92.44
1024	20.23	61.34	**92.49**
2048	74.01	60.36	92.44

Table E.13: Effect of MLP ratio *r*. We see that increasing the MLP ratio in general leads to better performance.

r	**#P**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**
1	2.77	60.98	92.45
2	3.82	61.31	92.48
3	4.87	61.02	**92.50**
4	5.92	**61.68**	92.44

Table E.14: Effect of depth of the ViTs *L*. We observe that increasing or decreasing the depth of the ViTs leads to poorer performance, with the best performance obtained when $L = 2$.

L	**#P**	**F2$_{\text{CIW}}$**	**F1$_{\text{Normal}}$**
1	3.82	61.05	92.51
2	5.92	**61.68**	92.44
3	8.02	60.53	**92.55**

6 Sinkhorn Tokenizer Cluster Visualizations

We visualize the cluster assignments within the Sinkhorn Tokenizer of the ResNet-50-MSHViT model to get a better understanding of how the non-local features are combined. For each cluster *k* we get the probability for each pixel that the pixel belongs to cluster *k*. We then visualize this map using a JET color mapping, where the mapping ranges from the minimum to maximum probability assignment. The JET color mapping maps the lowest value to blue and the largest value to red, with green as the intermediate color.

In tokenizers where there is information from previous scales, we visualize the clusters by first determining the assignment probability per pixel for the scale in focus. Then, for each cluster center from the previous scales we normalize the cluster assignments such that the maximum value is one. The cluster assignments are then multiplied by the assignment probability from the current scale cluster center and added to the overall assignment map. Lastly, the combined probability map is colored with a JET color mapping as before.

Examples are shown in Figure E.7. From these examples it is clear that not only does the Sinkhorn Tokenizer lead to non-local interactions, but captures the different scales of the defects. This is exemplified by the highlight of the multi-scale cracks as shown top example of Figure E.7 and the displaced pipe in the bottom example of Figure E.7. We observe that the clusters capture parts of the same regions, but within in different context such as one cluster center capturing a crack running along the pipe wall while another cluster center captures a cross section of the pipe.

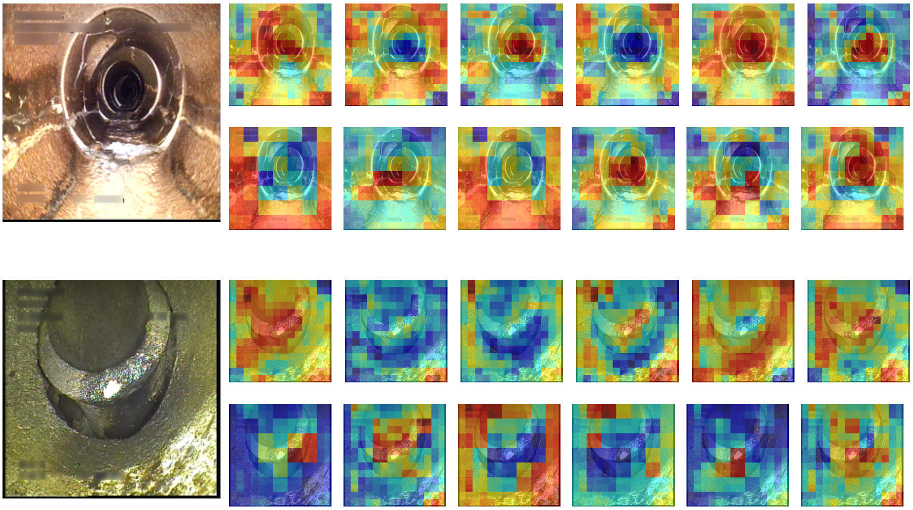


Fig. E.7: Visualization of the Sinkhorn Tokenizer clusters. We show a subset of the cluster assignments for two images using the ResNet-50-MSHViT model. The first image contains the classes **cracks, breaks, and collapses (RB)**, **displaced joint (FS)**, and **branch pipe (GR)**, and the second image contains the classes **surface damage (OB)**, **displaced joint (FS)**, and **connection with construction changes (OK)**. For each image, the top row contains examples of cluster assignment maps from the 4th scale clusters, while the bottom row contains examples of cluster assignment maps from the 5th scale clusters. See the description of the computation of the cluster assignment maps in Section 6.

7 Conclusion

Vision Transformers (ViTs) have taken the computer vision domain by storm, and led a surge in transformer focused research. A large part of this research focuses on exclusively using a transformer based architecture, while in comparison little attention has been given to the fusion of CNNs and transformers.

In this paper, we presented the Multi-Scale Hybrid Vision Transformer (MSHViT) for image classification, a natural extension of the hybrid vision transformer (HViT) which combines CNNs and ViTs, and the Sinkhorn Tokenizer, a clustering-based tokenizer based on Sinkhorn distances. The MSHViT extension enables the model to learn multi-scale non-local spatial semantics in the input, while the Sinkhorn tokenizer produces a smaller set of tokens that captures non-local spatial semantics.

We investigated the relative performance difference when extending ResNets with MSHViT and Sinkhorn tokenizer on the Sewer-ML multi-label sewer defect classification dataset, demonstrating a relative improvement of up to 2.53 percentage points. Through an extensive ablation study, we provided insights into the sensitivity of the introduced hyperparameters, verifying that the multi-scale extension outperforms regular HViTs, as well as qualitatively showing how the Sinkhorn tokenizer cluster centers captures distinct spatial semantics from one another.

Declarations

Financial interests. This research was funded by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project, and partially supported by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE), and by ICREA under the ICREA Academia programme.

Code availability. Code and model weights will be made available upon manuscript acceptance.

References

[1] American Society of Civil Engineers, "2017 infrastructure report card - wastewater," https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf, 2017, accessed: 20/3-2022.

[2] J. B. Haurum and T. B. Moeslund, "A survey on image-based automation of cctv and sset sewer inspections," *Automation in Construction*, vol. 111, p. 103061, 2020.

[3] C. H. Bahnsen, A. S. Johansen, M. P. Philipsen, J. W. Henriksen, K. Nasrollahi, and T. B. Moeslund, "3d sensors for sewer inspection: A quantitative review and analysis," *Sensors*, vol. 21, no. 7, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/7/2553

[4] Z. Liu and Y. Kleiner, "State of the art review of inspection technologies for condition assessment of water pipes," *Measurement*, vol. 46, no. 1, pp. 1 – 15, 2013.

[5] O. Duran, K. Althoefer, and L. D. Seneviratne, "State of the art in sensor technologies for sewer inspection," *IEEE Sensors Journal*, vol. 2, no. 2, pp. 73–81, April 2002.

[6] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 16 514–16 524.

[7] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[8] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7794–7803.

[9] H. Wang, Y. Zhu, B. Green, H. Adam, A. Yuille, and L.-C. Chen, “Axial-deeplab: Stand-alone axial-attention for panoptic segmentation,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 108–126.

[10] H. Zhao, J. Jia, and V. Koltun, “Exploring self-attention for image recognition,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 073–10 082.

[11] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 213–229.

[12] S. Liu, L. Zhang, X. Yang, H. Su, and J. Zhu, “Query2label: A simple transformer way to multi-label classification,” *CoRR*, 2021. [Online]. Available: https://arxiv.org/abs/2107.10834

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *ICLR*, 2021.

[14] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “Coatnet: Marrying convolution and attention for all data sizes,” *CoRR*, vol. abs/2106.04803, 2021. [Online]. Available: https://arxiv.org/abs/2106.04803

[15] K. Desai and J. Johnson, “Virtex: Learning visual representations from textual annotations,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 157–11 168.

[16] J. B. Haurum and T. B. Moeslund, “Sewer-ml: A multi-label sewer defect classification dataset and benchmark,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 451–13 462.

[17] M. Wang, H. Luo, and J. C. Cheng, “Towards an automated condition assessment framework of underground sewer pipes based on closed-circuit television (cctv) images,” *Tunnelling and Underground Space Technology*, vol. 110, p. 103840, 2021.

[18] M. Wang and J. C. P. Cheng, “A unified convolutional neural network integrated with conditional random field for pipe defect segmentation,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, pp. 162–177, 2020.

[19] Q. Zhou, Z. Situ, S. Teng, H. Liu, W. Chen, and G. Chen, “Automatic sewer defect detection and severity quantification based on pixel-level semantic segmentation,” *Tunnelling and Underground Space Technology*, vol. 123, p. 104403, 2022.

[20] L. M. Dang, H. Wang, Y. Li, T. N. Nguyen, and H. Moon, “Defecttr: End-to-end defect detection for sewage networks using a transformer,” *Construction and Building Materials*, vol. 325, p. 126584, 2022.

[21] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, “Deep learning based automated detection of sewer defects in cctv videos,” *Journal of Computing in Civil Engineering*, vol. 34, no. 1, p. 04019047, 2020.

[22] Y. Tan, R. Cai, J. Li, P. Chen, and M. Wang, “Automatic detection of sewer defects based on improved you only look once algorithm,” *Automation in Construction*, vol. 131, p. 103912, 2021.

[23] M. Wang, S. S. Kumar, and J. C. Cheng, “Automated sewer pipe defect tracking in cctv videos based on defect detection and metric learning,” *Automation in Construction*, vol. 121, p. 103438, 2021.

[24] Y. Li, H. Wang, L. Dang, M. Jalil Piran, and H. Moon, “A robust instance segmentation framework for underground sewer defect detection,” *Measurement*, vol. 190, p. 110727, 2022.

[25] J. B. Haurum, M. Madadi, S. Escalera, and T. B. Moeslund, “Multi-task classification of sewer pipe defects and properties using a cross-task graph neural network decoder,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

[26] J. B. Haurum, C. H. Bahnsen, M. Pedersen, and T. B. Moeslund, “Water level estimation in sewer pipes using deep convolutional neural networks,” *Water*, vol. 12, no. 12, 2020.

[27] F. Plana Rius, M. P. Philipsen, J. M. Mirats Tur, T. B. Moeslund, C. Angulo Bahón, and M. Casas, “Autoencoders for semi-supervised water level modeling in sewer pipes with sparse labeled data,” *Water*, vol. 14, no. 3, 2022.

[28] H. W. Ji, S. S. Yoo, B.-J. Lee, D. D. Koo, and J.-H. Kang, “Measurement of wastewater discharge in sewer pipes using image analysis,” *Water*, vol. 12, no. 6, 2020.

[29] H. W. Ji, S. S. Yoo, D. D. Koo, and J.-H. Kang, “Determination of internal elevation fluctuation from cctv footage of sanitary sewers using deep learning,” *Water*, vol. 13, no. 4, 2021.

[30] D. Ma, J. Liu, H. Fang, N. Wang, C. Zhang, Z. Li, and J. Dong, “A multi-defect detection system for sewer pipelines based on stylegan-sdm and fusion cnn,” *Construction and Building Materials*, vol. 312, p. 125385, 2021.

[31] Z. Situ, S. Teng, H. Liu, J. Luo, and Q. Zhou, “Automated sewer defects detection using style-based generative adversarial networks and fine-tuned well-known cnn classifier,” *IEEE Access*, vol. 9, pp. 59 498–59 507, 2021.

[32] C. Siu, M. Wang, and J. C. Cheng, "A framework for synthetic image generation and augmentation for improving automatic sewer pipe defect detection," *Automation in Construction*, vol. 137, p. 104213, 2022.

[33] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, "Generating synthetic point clouds of sewer networks: An initial investigation," in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis and P. Bourdot, Eds. Cham: Springer International Publishing, 2020, pp. 364–373.

[34] J. B. Haurum, M. M. J. Allahham., M. S. Lynge., K. S. Henriksen, I. A. Nikolov., and T. B. Moeslund., "Sewer defect classification using synthetic point clouds," in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP,*, INSTICC. SciTePress, 2021, pp. 891–900.

[35] Y.-W. Jeong, K.-B. Sim, S. Park, J. Oh, and W. J. Choi, "Generation of cnn architectures using the harmonic search algorithm and its application to classification of damaged sewer," *IEEE Access*, vol. 10, pp. 32 150–32 160, 2022.

[36] Y. Zhou, A. Ji, and L. Zhang, "Sewer defect detection from 3d point clouds using a transformer-based deep learning model," *Automation in Construction*, vol. 136, p. 104163, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092658052200036X

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[38] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*, vol. 139, July 2021, pp. 10 347–10 357.

[39] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021.

[40] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," *CoRR*, 2021.

[41] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 16 259–16 268.

[42] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 12 179–12 188.

[43] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *2021 IEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[44] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal self-attention for local-global interactions in vision transformers," *CoRR*, 2021.

[45] X. Cheng, H. Lin, X. Wu, F. Yang, D. Shen, Z. Wang, N. Shi, and H. Liu, "Mltr: Multi-label classification with transformer," *CoRR*, 2021.

[46] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 6824–6835.

[47] J. He, J. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, C. Wang, and A. Yuille, "Transfg: A transformer architecture for fine-grained recognition," *CoRR*, 2021.

[48] C.-F. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification," in *International Conference on Computer Vision (ICCV)*, 2021.

[49] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, and J. Gao, "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[50] J. Gu, H. Kwon, D. Wang, W. Ye, M. Li, Y.-H. Chen, L. Lai, V. Chandra, and D. Z. Pan, "Hrvit: Multi-scale high-resolution vision transformer," *CoRR*, 2021.

[51] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, 2013.

[52] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[53] Y. Zhu, Y. Zhu, J. Du, Y. Wang, Z. Ou, F. Feng, and J. Tang, "Make A long image short: Adaptive token length for vision transformers," *CoRR*, vol. abs/2112.01686, 2021. [Online]. Available: https://arxiv.org/abs/2112.01686

[54] S. Goyal, A. R. Choudhury, S. M. Raje, V. T. Chakaravarthy, Y. Sabharwal, and A. Verma, "Power-bert: Accelerating bert inference via progressive word-vector elimination," 2020.

[55] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, “Dynamicvit: Efficient vision transformers with dynamic token sparsification,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[56] D. Marin, J.-H. R. Chang, A. Ranjan, A. Prabhu, M. Rastegari, and O. Tuzel, “Token pooling in vision transformers,” *CoRR*, 2021.

[57] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *CoRR*, 2016.

[58] Y. M. Asano, C. Rupprecht, and A. Vedaldi, “Self-labelling via simultaneous clustering and representation learning,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Hyx-jyBFPr

[59] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 9912–9924.

[60] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[61] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[62] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI’17. AAAI Press, 2017, p. 4278–4284.

[63] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón, “Fnet: Mixing tokens with fourier transforms,” *CoRR*, 2021.

[64] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[65] T. Ridnik, E. Ben-Baruch, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, “Asymmetric loss for multi-label classification,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 82–91.

[66] T. Ridnik, H. Lawen, A. Noy, and I. Friedman, “Tresnet: High performance gpu-dedicated architecture,” *CoRR*, 2020.

[67] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7132–7141.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[69] W. Falcon, “Pytorch lightning,” https://github.com/PyTorchLightning/pytorch-lightning, 2019.

[70] R. Wightman, “Pytorch image models,” https://github.com/rwightman/pytorch-image-models, 2019.

[71] M. Dehghani, A. Arnab, L. Beyer, A. Vaswani, and Y. Tay, “The efficiency misnomer,” *CoRR*, vol. abs/2110.12894, 2021.

[72] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, “Perceiver: General perception with iterative attention,” *CoRR*, 2021.

[73] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, A. Brock, E. Shelhamer, O. Hénaff, M. M. Botvinick, A. Zisserman, O. Vinyals, and J. Carreira, “Perceiver io: A general architecture for structured inputs & outputs,” *CoRR*, 2021.

Part IV

Point Cloud-based Automation of Sewer Inspections

Paper F

Generating Synthetic Point Clouds of Sewer Networks: An Initial Investigation

Kasper Schøn Henriksen, Mathias S. Lynge, Mikkel D. B. Jeppesen, Moaaz M. J. Allahham, Ivan A. Nikolov, Joakim Bruslund Haurum and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 7th International Conference on Augmented Reality, Virtual Reality and Computer Graphics (SalentoAVR) pp. 364–373, 2020.

Abstract

Automatic robot inspections of sewer systems are progressively becoming more used for extending the lifetime of sewers and lowering the costs of maintenance. These automatic systems rely on machine learning and the acquisition of varied training data is therefore necessary. Capturing such data can be a costly and time consuming process. This paper proposes a system for generation and acquisition of synthetic training data from sewer systems. The system utilizes Structured Domain Randomization (SDR) for the generation of the sewer systems and an approximated model of a Pico Flexx Time-of-Flight camera for capturing depth and point cloud data from the generated sewer network. We evaluate the proposed system by comparing its output to ground truth data acquired from a Pico Flexx sensor in sewer pipes. We demonstrate that on average our system provides an absolute error of 5.78±8.92 and 7.58±8.68 mm, between data captured from real life and our proposed system, for two different scenarios. These results prove satisfactory for capturing training data. The code is publicly available at `https://bitbucket.org/aauvap/syntheticsewerpipes/src/master/`*.*

1 Introduction

The sewerage infrastructure is an essential part of modern society consisting of huge pipe networks hidden under the ground. This system is normally not a concern for the general public, but it is indispensable. To reduce maintenance costs, regular inspection of the entire sewer system is important. With today's methods this is not feasible, as inspection of sewer systems is a laborious task, carried out manually by skilled operators [1]. This is done by sending a remote-controlled platform with a camera through a section of the sewer system while an operator monitors the live video feed from this camera. As a result, pipes are often replaced prematurely to avoid older pipes causing issues. Pipes with an expected lifetime of 75 years are replaced after just 50 years, causing several hundred million DKK in expenses for the Danish state [2]. To improve this lack of inspection, an ongoing research project, the Automated Sewer Inspection Robot (ASIR) project, intends to develop an autonomous robot, that will roam the sewers and utilise machine learning to identify defects [2]. For a machine learning algorithm to properly identify defects and landmarks such as joints and branch pipes in a sewer network, the algorithm that is utilised by the robot requires training, and for this, an annotated data set is desired. The intended plan of collecting data from the sewers is using depth- and RGB cameras. Currently, only databases with RGB images from CCTV inceptions are available, and a data set consisting of depth information is needed. As capturing data from real sewers can be a huge and expensive task, an alternative could be generating synthetic data.

Different types of depth sensors can be utilised for capturing depth information, such as Time of Flight (ToF) [3, 4], stereo cameras [5], and structured light [6]. The ToF sensor Pico Flexx by Camboard is chosen as the sensor to be implemented as part of the proposed system. The sensor is chosen because of its small size, high resolution and low power consumption. It is also shown by [7] that ToF sensors are ideal for use in inspection of hazardous and hard to reach places. The Pico Flexx outputs point clouds, which offers more depth information that can help identify defects in the sewer network. This sensor will be implemented and will be the focus of this paper.

Contributions: Our contribution is a novel system, making the first foray into generating Synthetic Point Clouds (SPC) in the sewer domain. This system consists of two parts;

- Generate virtual sewer networks which are based on the method Structured Domain Randomization (SDR) method.
- Generate SPCs, approximating the Pico Flexx output, that can be used for training machine learning algorithms.

2 Related Work

We will be looking at the state of the art for both the sewer generation and ToF simulation.

Depth sensors: Bulczak *et al.* [8] suggests a method for simulating amplitude modulated continuous wave (AMCW) ToF sensors including the artefacts caused by multipath interference (MPI). MPI is of particular interest for simulating sewer inspection in textured pipes, which could be of interest even in smoother plastic pipes, as it could model reflections. This method increases computational costs, however, it is specifically designed for GPU execution to allow fast processing.
Sarbolandi *et al.* [9] compares structured light and ToF depth sensing in detail, specifically comparing the two different versions of the Microsoft Kinect sensors that employ these different technologies. They found that the ToF variant performed better at rejecting background illumination, motion, highly translucent objects, and at large incident angles.
Sarker *et al.* [10] explores the use of a stereo vision based depth camera for the use of crack detection in concrete, showing the ability to detect and classify variants of cracks.

Synthetic generation of environments: In recent years the interest in generating annotated synthetic data has been steadily increasing, as to meet the demands of more complex deep learning architectures in an inexpensive manner [11–14]. Tobin *et al.* [15] presents a method for computer generation of synthetic data, called Domain Randomization (DR), which is achieved by generating structured variations of a chosen scenario, such that a machine learning algorithm would process real images as if they

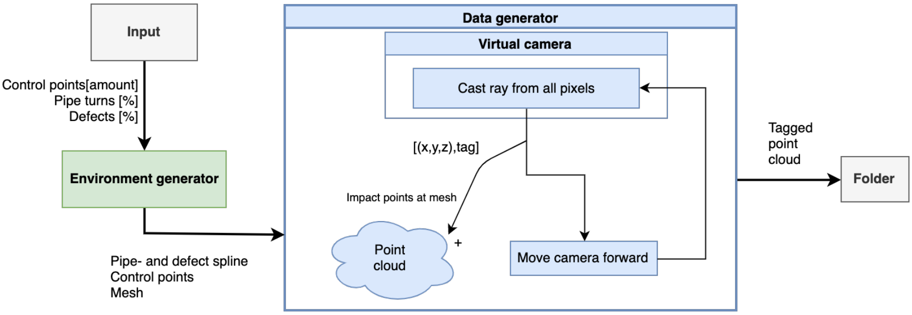


Fig. F.1: Structure of the proposed solution.

were another variation of the scenario. The work was further developed by Prakash *et al*. [16], into an expanded version of DR: Structured Domain Randomisation (SDR). Instead of placing objects by a uniform random distribution with DR, context of the scene is considered. This structures the randomisation by a set of rules that corresponds to the intended environment. This allows machine learning algorithms to train not only on the virtual objects placed in the scene, but also on the context of where these objects are positioned in relation to each other. The SDR method is found to fit the focus of our solution, and it will be used for generating the environment that will later be used for the 3D data acquisition.

3 Methods and Material

The proposed solution is implemented in Unity, which consists of two parts: *Environment Generator* and *Data Generator*. The overall structure of the system is shown in Figure F.1.

The *Environment Generator* dynamically generates a virtual sewer network consisting of splines, control points and pipe meshes based on given parameters. These are passed to the *Data Generator* that generates point clouds based on them and two additional parameters: folder path and number of point clouds. Folder path: defines the output folder for point cloud data. Number of point clouds: defines how many point clouds to create. The camera moves forward until this parameter is satisfied.

3.1 Pico Flexx

Camboard Pico Flexx, is a joint production by PMD Technologies and Infineon. It is based on PMD ToF technology (AMCW-ToF), which uses Near Infrared (NIR) laser to determine the distance from the sensor to the impact points of objects. The small size and the low power consumption enhances the number of possible applications. To mimic the Pico Flexx, its parameters should be implemented for the synthetic sensor. Because the information available from PMD [17] only shows camera characteristics such as field of view and aspect ratio, a complex ToF simulation can be difficult to be implemented as more information is required. The sensor approximation in this paper is therefore estimated using ray casting.

3.2 Environment Generator

The gap between the real- and the virtual domain can be reduced using 3D models that mimic physical models and utilize SDR to generate the environment. 3D models and environment generator will be explored in the upcoming sections.
Pipes & Defects: To mimic the physical environment, 3D representations of physical pipes and defects are utilised. The system generates points along the network, where handmade defects can spawn.
Generator: To enhance the generator, the structure of how objects are generated should be acknowledged by utilizing SDR. As the sewer system domain differs to the domain in the SDR paper, other contexts such as rubber rings mostly occurring at pipe displacements should be considered. SDR proposes a taxonomy which covers the following four principles. *Scenario:* determines general parameters to generate the domain such as length of the sewer network and defect probability. *Global Parameters:* generates contextual splines based on parameters from the scenario. *Context Splines:* instantiate objects based on given probabilities and context. *Objects:* contain a transform, 3D mesh, collider, and defect- or fine tag.

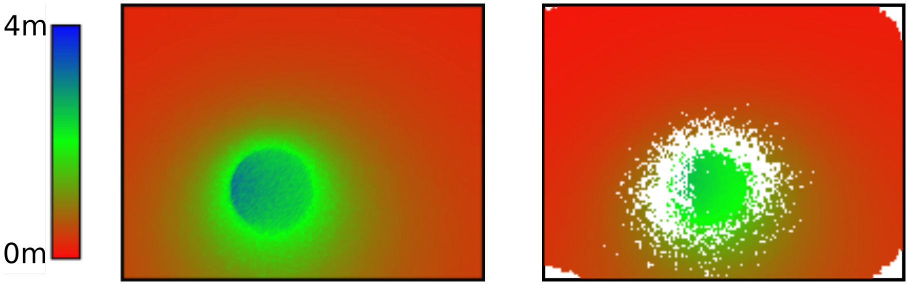
3.3 Data Generator

Without extensive information about the Pico Flexx, an approximated virtual camera in Unity is set up to mimic the output of the Pico Flexx. The virtual camera utilises the same resolution and focal length. Moreover, to detect depth in a scene, rays are cast up to the distance of the detectable depth of the Pico Flexx. These rays are cast from each pixel of the virtual camera in the direction of the viewport in normalized coordinates. Figure F.2 shows a comparison of 2D depth images from the data generator and Pico Flexx. However, to evaluate the data generation, comparing point clouds is preferred to avoid dimensionality reduction.

In a simulated environment, a physics engine can utilise ray casting to determine impact positions at 3D meshes. This provides additional information for each ray which can label the points within the point cloud. For each ray, random noise from a Gaussian distribution is added with a range of $\pm 1\%$ from the true value, based on the datasheet from PMD [17].

4 Results

To evaluate the solution, point clouds from both Pico Flexx and the system using similar setups are acquired.



(a) Synthetic

(b) Pico Flexx

Fig. F.2: Output image from the image generator, compared to the Pico Flexx, where white indicates missing points.

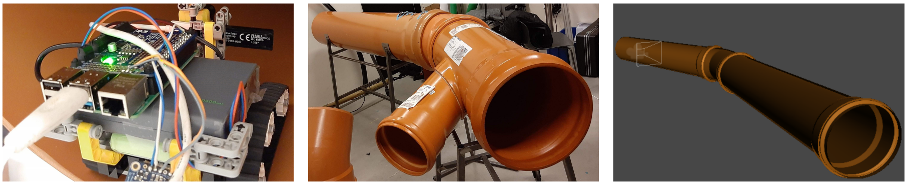
4.1 Data Gathering from Pico Flexx

To gather Pico Flexx data, a controlled environment was set up, and an already built robot with a mounted Pico Flexx was used, as shown in Figure F.3a. It is preferred to get data around pipe connections, as displaced pipes are the most common defect[1]. Physical pipes were set up inside a windowless room, to avoid external light sources disturbing the Pico Flexx sensor that was attached to a small remote controlled mobile platform. Based on the setup in Figure F.3b, two scenarios were arranged; the first setup without displacement and second with displacement. In the first scenario (**S1**), the robot was placed at the start of the pipe and programmed to move forward through the pipe. In the second scenario (**S2**), the robot was placed 50 cm from both the first pipe connection and a misplaced rubber ring, and programmed to move through the connection. The outcome from these scenarios was point cloud data sequences split by time stamps. In order to evaluate the Pico Flexx data, the sewer system and scenarios of the physical setup were mimicked for the virtual setup, as shown in Figure F.3c where the Pico Flexx's orientations were mimicked by approximately transforming the virtual camera. Using these virtual scenarios, SPCs were extracted to be compared to the Physical Point Clouds (PPC).

4.2 Point Cloud Comparison

To evaluate the simulation performance, differences and similarities between the PPCs and SPCs were compared, which allows a per point distance calculation. For this, the widely used software CloudCompare [18], which has been tested by several studies [19, 20], is utilised. The PPC will be used as the ground truth whilst the SPC will be used to compare with. Before comparing the point clouds, they are aligned using the fine registration method, Iterative Closest Point (ICP) [21], which aligns the desired point clouds to the ground truth, by minimizing the distance error between them. As

[1] Based on unpublished works



(a) The robot **(b)** Physical setup **(c)** Virtual setup

Fig. F.3: The figures represent the robot's placement, the physical- and virtual setup, respectively. Note: The branch pipe is not within the sensor's FoV.

missing points exist in the PPCs, local modelling is utilised to approximate planes based on least square to estimate the pipe geometry. This minimizes the distance between point clouds, as approximated planes can cover holes in the PPCs.

The absolute distances distribution of the compared point clouds for the two scenarios is presented in Figure F.5. The results show a difference of 5.78±8.92 mm for **S1** and 7.58±8.68 mm for **S2**.

4.3 Accuracy of Point Clouds

To ensure both the synthetic- and physical data give correct measurements of the real pipe, the diameter error is calculated. This is done on **S1** as this is the scenario with the longest straight section. Each point cloud is sliced into overlapping segments of 100 mm, using a step size of 10 mm. At each segment, the distance from each point to its most distant point is calculated, and the mean of these distances are then used as the diameter for that section.

5 Discussion

In the point cloud comparison, some of the error described by the mean in both scenarios, may be caused by a difference in placement of the Pico Flexx and the virtual camera, as the placement of the virtual camera was done by hand, as close to the Pico Flexx positioning as possible. As seen in Figure F.5, outliers are present, with the largest occurrence is in **S1**. The outliers appear at the pipe connection of the pipes which can be seen in Figure F.4, by the yellow and red colors. The remaining outliers in **S1** are located at the end of the point cloud, likely caused by the lower point density. In **S2**, the outliers are most noticeable around the pipe connection point, but in this case they seem to be caused by the shadow cast by the rubber ring in **S2**. This can be seen in Figure F.4 as a lack of points. Considering the mean for all evaluations, in regards to the size of the pipes, the error rate is considered acceptable in order to classify *e.g.* displacements.

Figure F.6 shows the diameter size along the point clouds from the accuracy of

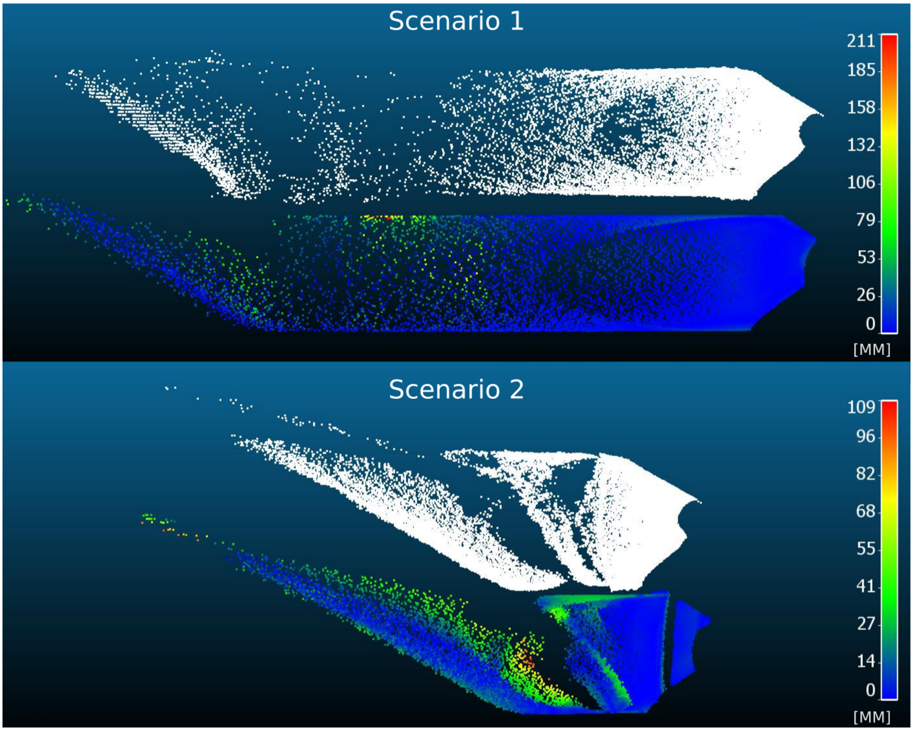


Fig. F.4: This figure represents the two scenarios, where the white point clouds are PPCs and the colored point clouds are SPCs. Moreover, the legend indicates the absolute distance error value for the gradient colors in the SPCs.

point cloud comparison. The first 25 cm section of both the point clouds shows a diameter of 0 mm, and that is due to the distance between the sensor and the pipe surface. The next 35 cm section has increasing diameters size due to the slanted shape of the point cloud at the beginning as can be seen in Figure F.4, which results in a non-circular slice. From 60 until 150 cm, the mean diameter is 405mm for the PPC and 376mm for the SPC, which shows a small error in the PPC, believed to be caused by the lack of points in that area of the point cloud. However, the SPC shows almost a diameter size that matches the real pipe diameter for the majority of the pipe length. After 160 cm, major errors occur in both point clouds, which starts around the bend of the pipe, this is not of interest as the test assumes the pipe is straight. The point cloud comparison test indicates that the Pico Flexx yields more noise than expected, but to clarify this, further research is required. The accuracy of the point clouds test, indicates that the imperfections can be caused by the textures, imperfectness of the surfaces, dirt in the pipes or lighting that could reflect within the pipes' internal surfaces. All these characteristics define the appearance of real surfaces, which can have an impact on other elements when implemented in computer graphics.

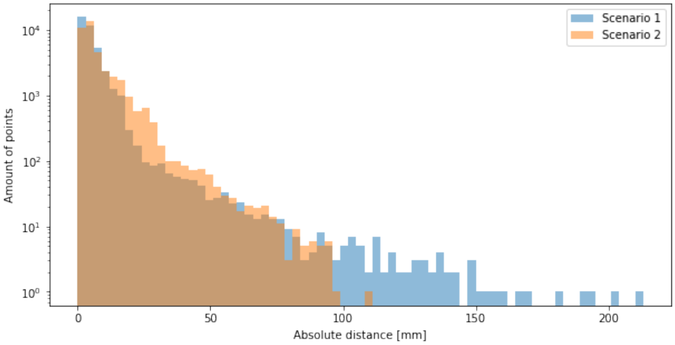


Fig. F.5: Histogram of the absolute distances from points in the SPC, to the corresponding calculated least square plane in the PPC. Note the y-axis is log-scaled. Scenario 1 mean: 5.78±8.92 mm. Scenario 2 mean: 7.58±8.68 mm.

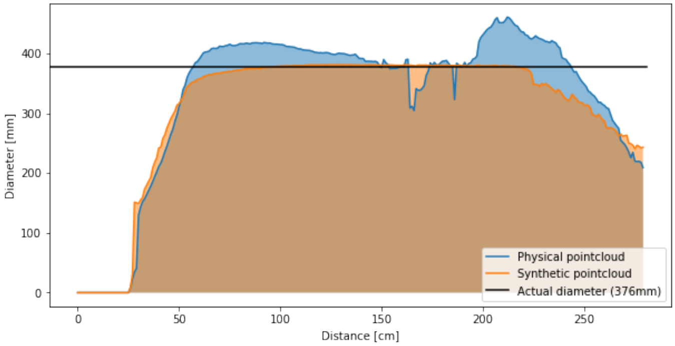


Fig. F.6: Error rate compared to the specified diameter of the real pipe from **S1**.

6 Conclusion

This paper has introduced a system which is capable of generating annotated SPCs based on the characteristics of the Pico Flexx in a synthetic sewer environment. Through experiments, it is concluded that the difference of the two point clouds is mainly caused by the lack of points in areas of the PPC. Furthermore, it is found that regions of the point clouds have points that approximately match the real data. Using a moving robot, a continuous good point cloud could be stitched together from multiple instances of the good section. However, if a larger portion of the point cloud is to be used, a more accurate simulation of the sensor might be required, for example an implementation of the MPI simulation as mentioned in related work section 2. Considering the simplicity of the calculation behind the SPCs, the result seems promising. This especially applies in relation to the first part of the point clouds in **S1**. It should be noted that this is a controlled environment, and entering more realistic scenarios, that contain water, dirt etc. might challenge the Pico Flexx and therefore it yields noisier data. Here the solution might perform poorly due to it being too idealised, therefore reflection, refraction etc. need to be taken into considerations. Overall the solution is able to generate randomised sewer systems that contain defects using SDR. Moreover, SPCs can be generated which can potentially be used to accelerate the data acquisition process for machine learning algorithms.

Acknowledgment: This work is supported by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project. The authors declare no conflict of interests.

References

[1] DTVK, *TV-H, Håndbog for TV-inspektion*, 2019.

[2] EnviDan, "Envidan er med i nyt udviklingsprojekt, hvor robotter i kloakken vil spare samfundet for kæmpe millionbeløb!" 2019. [Online]. Available: https://www.envidan.dk/cases/asir-udviklingsprojekt

[3] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor - system description, issues and solutions," in *2004 Conference on Computer Vision and Pattern Recognition Workshop*, June 2004, pp. 35–35.

[4] R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *IEEE Journal of Quantum Electronics*, vol. 37, no. 3, pp. 390–397, March 2001.

[5] R. A. Hamzah and H. Ibrahim, "Literature survey on stereo vision disparity map algorithms," *Journal of Sensors*, vol. 2016, pp. 1–23, 2016.

[6] R. Valkenburg and A. McIvor, "Accurate 3d measurement using a structured light system," *Image and Vision Computing*, vol. 16, no. 2, pp. 99–110, 1998.

[7] R. M. Jans, A. S. Green, and L. J. Koerner, "Characterization of a miniaturized ir depth sensor with a programmable region-of-interest that enables hazard mapping applications," *IEEE Sensors Journal*, pp. 1–1, 2020.

[8] D. Bulczak, M. Lambers, and A. Kolb, "Quantified, interactive simulation of amcw tof camera including multipath effects," *Sensors*, vol. 18, no. 1, p. DOI: 10.3390/s18010013, 2018.

[9] H. Sarbolandi, D. Lefloch, and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect," *CoRR*, 2015.

[10] M. Sarker, T. Ali, A. Abdelfatah, S. Yehia, and A. Elaksher, "A cost-effective method for crack detection and measurement on concrete surface," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, p. 237, 2017.

[11] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3234–3243.

[12] A. Kamilaris, C. van den Brink, and S. Karatsiolis, “Training deep learning models via synthetic data: Application in unmanned aerial vehicles,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2019, pp. 81–90.

[13] T. Bruls, H. Porav, L. Kunze, and P. Newman, “Generating all the roads to rome: Road layout randomization for improved road marking segmentation,” in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019, pp. 831–838.

[14] J. Fang, D. Zhou, F. Yan, T. Zhao, F. Zhang, Y. Ma, L. Wang, and R. Yang, “Augmented lidar simulator for autonomous driving,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1931–1938, April 2020.

[15] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.

[16] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, “Structured domain randomization: Bridging the reality gap by context-aware synthetic data,” *CoRR*, 2018, accessed: 2020-02-28.

[17] pmd. (accessed: 13/12-2019) Development kit brief camboard pico flexx. [Online]. Available: https://pmdtec.com/picofamily/wp-content/uploads/2018/03/PMD_DevKit_Brief_CB_pico_flexx_CE_V0218-1.pdf

[18] ENST, EDF, Daniel Girardeau-Montaut, “Cloud compare,” accessed: 01/02-2020. [Online]. Available: http://www.cloudcompare.org/

[19] Y. Rajendra, S. Mehrotra, K. Kale, R. Manza, R. Dhumal, A. Nagne, and A. Vibhute, “Evaluation of partially overlapping 3d point cloud’s registration by using icp variant and cloudcompare,” 2014.

[20] E. Oniga, A. SAVU, and A. Negrila, “The evaluation of cloudcompare software in the process of tls point clouds registration,” *RevCAD Journal of Geodesy and Cadastre*, vol. 21, pp. 117–124, 12 2016.

[21] S. Rusinkiewicz and M. Levoy, “Efficient variants of the icp algorithm,” 2001.

Paper G

Sewer Defect Classification using Synthetic Point Clouds

Joakim Bruslund Haurum, Moaaz M. J. Allahham, Mathias S. Lynge,
Kasper Schøn Henriksen, Ivan A. Nikolov and Thomas B. Moeslund

The paper has been published in the
Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications pp. 891–900, 2021.

The layout has been revised.

Abstract

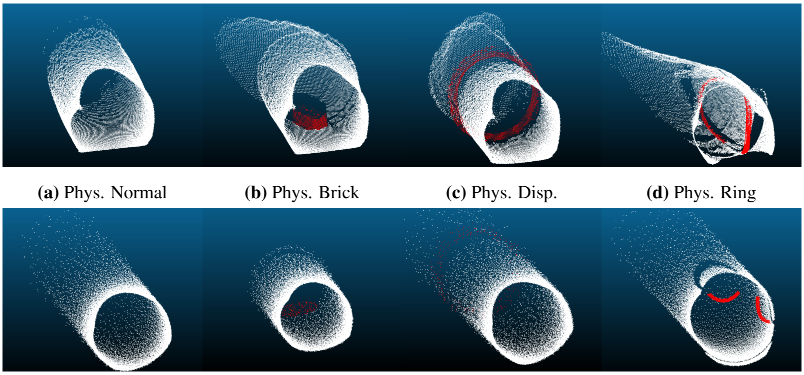
Sewer pipes are currently manually inspected by trained inspectors, making the process prone to human errors, which can be potentially critical. There is therefore a great research and industry interest in automating the sewer inspection process. Previous research have been focused on working with 2D image data, similar to how inspections are currently conducted. There is, however, a clear potential for utilizing recent advances within 3D computer vision for this task. In this paper we investigate the feasibility of applying two modern deep learning methods, DGCNN and PointNet, on a new publicly available sewer point cloud dataset. As point cloud data from real sewers is scarce, we investigate using synthetic data to bootstrap the training process. We investigate four data scenarios, and find that training on synthetic data and fine-tune on real data gives the best results, increasing the metrics by 6-10 percentage points for the best model. Data and code is available at `https://bitbucket.org/aauvap/sewer3dclassification`.

1 Introduction

The sewerage infrastructure is one of the largest, but also most forgotten, infrastructures in our modern society. In the United States there are currently approximately 2 million km of sewer pipes serving nearly 240 million Americans. By 2036 the sewerage infrastructure is expected to serve an additional 56 million users [1]. The size of the sewerage infrastructure poses a clear problem during maintenance, as it is near impossible to regularly inspect all stretches of sewer pipes. Furthermore, sewer maintenance requires skilled inspectors who are capable of operating the required equipment to inspect the buried pipes. These inspections are conducted using a remote-controlled "tractor", which the inspector controls from a vehicle above ground. This can be both demanding and slow, and potentially prone to human errors.

To deal with this problem, one possibility is to use an autonomous or semi-autonomous robotic solution. Such solutions have been successfully developed and deployed for tunnel walls inspection [2], transmission and electrical wires [3], underwater ship hulls [4], wind turbine blades [5], among others. An important characteristic that each of these solutions share, is that the robotic system needs to have appropriate sensors for both self-localization and mapping the environment, as well as capturing enough information from the surfaces such that a proper inspection of potential damages or obstructions can be achieved. To ensure that enough information is captured, 3D information in the form of depth images and point clouds, is chosen in addition to traditional 2D images. To capture such information, different sensor can be used - LiDAR laser scanners [6, 7], stereo cameras [8], photogrammetry [9], time-of-flight and structured light cameras [10, 11].

Sewer inspection data presented in the state-of-the-art is normally not available as public datasets, and the ones used are focused around 2D RGB images [12]. How-



(a) Phys. Normal **(b)** Phys. Brick **(c)** Phys. Disp. **(d)** Phys. Ring

(e) Synth. Normal **(f)** Synth. Brick **(g)** Synth. Disp. **(h)** Synth. Ring

Fig. G.1: Example point clouds from the real and synthetic pipe setup. Defects are shown in red for easier visualization.

ever, capturing large amounts of 3D inspection data from sewers is not a trivial task. Therefore, we look into using synthetic data for training a sewer inspection algorithm. The creation of such synthetic data has been detailed in the work of [13], where sewer pipes were 3D modeled and used in a custom simulation environment, together with an approximated PMD Pico Flexx [14] time-of-flight camera, to generate 3D point clouds. We therefore look into using synthetic data to bootstrap the training process of a deep learning based 3D sewer defect classifier. The main contributions of this paper are threefold:

1. A publicly available dataset of synthetic and real point clouds of normal and defective sewer pipes.
2. Demonstrating the feasibility of using 3D point clouds and geometric deep learning methods for classifying sewer defects.
3. A comparison of the effect of synthetic and/or real data when training a defect classifier.

2 Related Work

Automated Sewer Inspections. Vision-based automation of sewer defects has traditionally been based on 2D image data from Closed-Circuit Television (CCTV) and Sewer Scanner and Evaluation Technology (SSET) sewer inspections. CCTV and SSET inspection data have been used for nearly 30 years, with methods ranging from morphology based discriminators [15–18], to using feature descriptors and machine learning classifiers [19–21], and within the recent years using deep learning for classification, detection, and segmentation [22–25]. For an in-depth review of these methods we refer to the survey by [12]. There has, however, been significantly less work on detecting defects using 3D sensors. 3D sensors are interesting as some sewer defects, such as displaced joints and obstacles, may not be immediately visually apparent, but can be obvious when looking at the depth information. Traditionally two types of sensors have been used: Laser scanners and ultrasound. Laser scanners have been used extensively by Duran *et al.* for binary defect detection of cracks, defective joints, and obstacles, by utilizing depth and the intensity of the reflected light as input for fully-connected neural networks [26–28]. Similarly, [29] designed a novel laser scanner for detecting displaced joints, cracks, and deposits, which works in a comparable way as to CCTV inspections. [30] similarly proposed a novel laser scanner design for defect detection and extracting pipe geometry. Furthermore, [31] and [32] have used laser scanners for navigation purposes as well as detecting defects and recovering the geometry of the pipe. [33] utilized ultrasound based methods for detecting cracks and holes in concrete pipes. Khan and Patil have proposed detection cracks in PVC pipes by analyzing the acoustic response under different conditions using frequency domain analysis [34, 35]. Alejo *et al.* have utilized RGB-D camera for localization and defect classification, utilizing graph based learning and convolutional neural networks (CNN) [36, 37]. Furthermore, as documented by [12] there is a lack of public dataset and code releases for methods based on CCTV and SSET inspections, which is also the case with methods designed for inspections using 3D sensors.

Geometric Deep Learning. Within recent years the application of deep learning methods on unstructured 3D data, such as point clouds, have gained interest within the computer vision community. The earliest methods utilized specialized voxel-based methods [38] and reutilizing 2D CNNs in a multiview-based approach [39] in order to classify objects, resulting in, respectively, high memory consumption and slow computation times. Qi *et al.* were the first to successfully process the raw point clouds using the fully-connected neural network architectures, PointNet [40] and PointNet++ [41]. This work has been expanded upon within the autonomous vehicle community for object detection and segmentation [42, 43], amongst other point based methods [44, 45]. 3D point clouds can also be observed as a graph problem, which was utilized by [46] in the Dynamic Graph CNN (DGCNN) architecture, where edge information between points are aggregated to better learn local and global information. For a review of the geometric deep learning field we refer to the work of [47] and [48].

Synthetic Data. In the current era of machine learning based methods, representative training data is essential. However, it may not always be possible to acquire the necessary training data, as it can be prohibitively expensive. This is especially apparent when working on tasks where the interesting parts are *rare*, such as defect detection. The generation of representative synthetic data has therefore been increasingly investigated. [49] proposed the Domain Randomization (DR) method, which generates randomized renderings of a scene in order to train a robot. [50] expanded on this method by accounting for the structure in the scene, called Structured Domain Randomization (SDR), which was demonstrated on the KITTY object detection task. [51] showed using

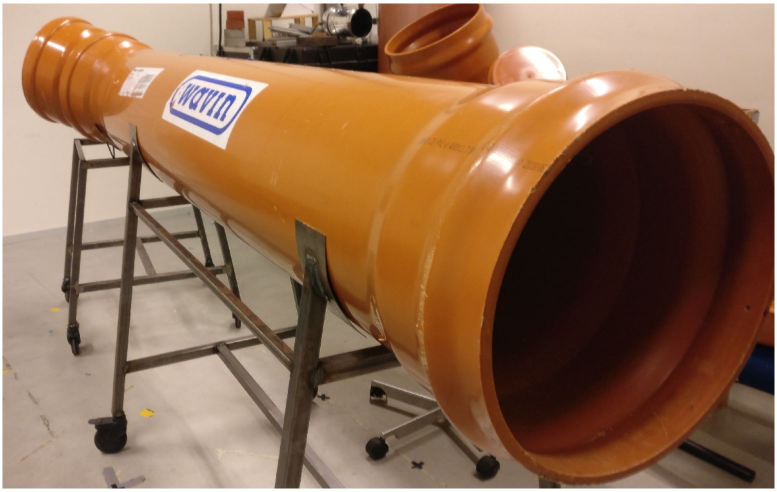


Fig. G.2: An example pipe configuration, used for collecting the point cloud data from the physical setup.

large amount of synthetic data can help handle the long tailed distribution that occurs in the animal classification task, showing the promise of synthetic data. Lastly, [13] proposed an SDR based synthetic data generator for PVC sewer pipes, which can generate displaced joints and defective rubber rings in the joints.

3 Dataset

As mentioned in Section 2 there are currently no publicly available datasets within the sewer inspection field. We therefore construct our own dataset, consisting of normal non-defective pipes and defective pipes with three different kinds of defects: displaced joints, defective rubber rings, and obstructions in the form of bricks. The three defect types are selected as they are observed frequently in the real world. As 3D sensors are very rarely used for sewer inspections, the constructed dataset consists of synthetic point cloud data, as well as real data obtained in a lab environment.

3.1 Synthetic Data Generation

We base our synthetic data generation on the SDR-based approach proposed by [13]. The proposed data generator generates a random sewer network consisting of clean PVC pipes, with no water or sediments, and randomly places defects along the pipes. A virtual approximation of the PMD Pico Flexx time-of-flight sensor is moved through the sewer network, and record synthetic point clouds.

The generated defects are, however, constrained to only displaced joints and defective rubber rings, which are concurrent. We update the simulator to allow displaced joints and defective rubber rings to occur independent of each other, and further extend

Table G.1: Overview of the data in the different data splits. The Displacement (Disp.), Brick, and Rubber Ring columns represent the amount of point clouds for the three investigated defect types.

	Synthetic				Real				
Split	**Normal**	**Disp.**	**Brick**	**Rubber Ring**	**Normal**	**Disp.**	**Brick**	**Rubber Ring**	**Total**
Training	5,365	1,811	1,822	1,802	140	45	45	44	11,074
Validation	1,385	439	428	448	31	12	12	13	2,768
Test	1,350	450	450	450	244	85	76	80	3,185
Total	8,100	2,700	2,700	2,700	415	142	133	137	17,027

it to allow for randomly placed bricks in the pipe. Bricks are chosen, because of their relatively basic shape, not prone to many variations, compared to other possible obstructions in sewer pipes. This way the overall defect classification performance of the algorithms can be evaluated, without the need to create too many different shape cases. The bricks are placed by applying a random force to the brick, which pushes it into a random position and orientation in the pipe. We constrain the simulator to only allow one kind of defect per extracted point cloud, in order to be able to determine the effect of each type of defect.

3.2 Physical Data Collection

In order to collect point clouds from a set of real PVC sewer pipes, a physical setup was created in an indoor laboratory, see Figure G.2. The data was collected using a PMD Pico Flex sensor. As no sewer data captured with the Pico Flex sensor is available, we conduct a simple test, to verify its accuracy presented in its datasheet [14]. The sensor is mounted on a moving platform and directed towards a white wall with an approximately Lambertian surface. The sensor is then moved away from the wall at equal 0.1m intervals, starting from 0.2m until 2m. A Leica DISTO laser range finder, is used to capture ground truth data at each position, as it has a known accuracy of 0.03m. The two sensors are calibrated to the same distance measurement at 0.2m. The difference between the two are presented on Figure G.3. The distance errors are higher than the ones given in the datasheet [14] for the camera. This needs to be taken into account, as these distance errors, might result in noise or deformations in the selected pipe segments, especially between 0.8m and 1.5m.

Five different pipe segments, with a diameter of 400 mm were used for data collection: two straight pipes, and three corner pipes with turning angles of 15, 30, and 45 degrees. The pipe segments were combined in different permutation, with the sensor moved through the pipes while placed in the center. Defects were added to the pipes by randomly placing bricks or rubber rings in the pipes, or displacing the joints of the pipe segments. As in the synthetic data generator, only one type of defect is present at a time.

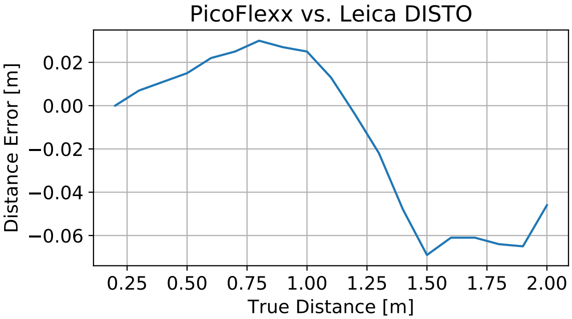


Fig. G.3: The Pico Flexx distance errors, compared to a laser range finder at distances between 0.2m and 2m.

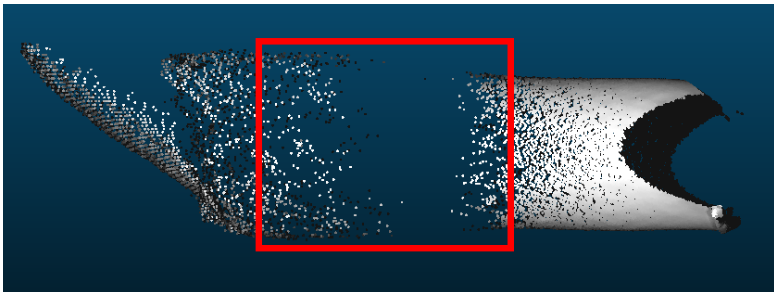


Fig. G.4: Example of point clouds captured with the real Pico Flexx and the holes, caused by missing data.

3.3 Comparison Between Real and Synthetic Data

Examples of the real and synthetic data are shown in Figure G.1, with one example per class. One problem found from the real data captured with the Pico Flexx sensor, is the presence of "holes" in both the depth map and the point cloud - areas, where no depth data is captured. These holes depend on the environmental lighting, the distance and orientation of the imaged surface, compared to the camera, as well as the glossiness of the surface. Examples of such holes can be seen in Figure G.4. One way we address this problem is by subsampling both the synthetic and real point cloud data, which lowers the density variation of the point clouds. More information, about the subsampling process can be found in Section 4.1.

3.4 Dataset Split

The acquired synthetic and real data are divided into training, validation and test splits, as shown in Table G.1. We choose to place the majority of the real data (85%) in the test split, as to reflect the real world data situation, where inspection data is in the form of CCTV and SSET videos and annotated 3D data is limited. Inversely, we utilize the majority of the synthetic data in the training and validation splits. We make sure there is no data leakage between splits by generating new synthetic data for each split, and splitting the real data based on the pipe segment configurations. We balance the amount of defective and normal data, such that the problem is more well-behaved, which is standard within the sewer defect classification field [22, 23].

4 Methods

The proposed method consists of two steps: preprocessing the data and the deep learning models.

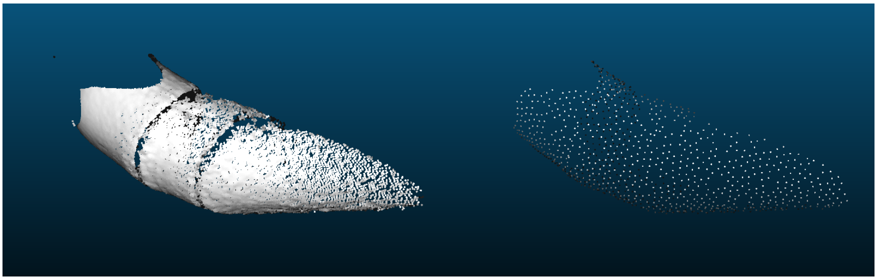
4.1 Data Preprocessing

Training a deep learning model on the raw point cloud data is infeasible due to the large number of points, leading to high memory consumption. It is therefore necessary to subsample the point clouds in order to efficiently process them. Before subsampling point clouds, it is preferred to reduce the number of outliers that may occur. This can prevent subsampling approaches being biased by the outliers and rather focus on points containing relevant geometric information of a given pipe. Points that are stored in the origin of a point cloud are discarded, as they represent points that did not return a valid value. Afterwards, Statistical Outlier Removal (SOR) [52] is applied to discard aberrative points that heavily differ from the geometric representation of a pipe.

We subsample the point clouds to 1024 points, the number of points originally used for the PointNet approach. Traditionally the subsampling step has been performed by applying the Farthest Point Sampling method, which iteratively selects the point in the point cloud which is farthest away from the previously selected points [40]. This is, however, not the best approach for our data, as some defects manifest themselves as points in the middle of the pipe, which would be subsequently removed. Therefore we apply two different subsampling approaches sequentially. First we apply a spatial subsampling step [53], which enforces a minimum distance, d, between each point. d is selected such that more than 1024 points remain, though d may change per point cloud. d is initially set to 0.03, and decremented by 0.004 each time the resulting point cloud has less than 1024 points. Afterwards the point cloud is reduced to 1024 through uniformly sampling the subsampled point cloud. As a last step the subsampled point clouds are normalized into a unit sphere. Examples of a pipe segment before and after the preprocessing steps can be seen in Figure G.5.

4.2 Model Architectures

We investigate the performance of two state-of-the-art geometric deep learning methods: PointNet [40] and DGCNN [46]. We choose PointNet to get a baseline performance, whereas DGCNN is chosen to evaluate the effectiveness of the advances within the field. PointNet is built upon sequentially applying the same fully-connected sub-networks on



(a) Before Subsampling **(b)** After Subsampling

Fig. G.5: Example of a sewer pipe segment before and after the subsampling preprocessing steps

the individual points, in parallel. This way each point is processed independently of any other points. In order to aggregate the feature information of each point, the symmetric max pooling function is used. Furthermore, the PointNet architecture includes a special sub-network called a T-Net, which predicts an affine transformation matrix used to align the input into a canonical form. The T-Net is applied in the beginning on the raw input, as well as the intermediate features. However, the intermediate feature alignment matrix is learned in a high dimensional space, which makes the optimization process more difficult [40]. Therefore, [40] regularize the feature alignment matrix, A, by forcing it to be close to an orthogonal matrix, as shown in Equation G.1.

$$L_{reg} = ||I - AA^T||_F^2 \tag{G.1}$$

The DGCNN network builds upon the PointNet architecture, by introducing the EdgeConv layer between each of the shared fully-connected subnetworks. For each point, x_i, in point cloud, the EdgeConv layer finds the k closest points in the feature space, x_j, including the point itself. For all k points, a learnable edge function, denoted $h(x_i, x_j)$, is applied, and the obtained edge features are aggregated using a symmetric aggregation function. In DGCNN, h is defined as a fully-connected network which takes the concatenation of x_i and $x_j - x_i$ as input, while the aggregation function is a simple channel wise max operation. This way both global and local shape information is captured in the EdgeConv layer.

5 Experimental Results

We approach the task as a multi-class classification task, where we have to determine whether the point cloud represents a sewer with one of the three considered defects, or whether it is a normal sewer pipe. The PointNet and DGCNN networks are trained and evaluated using the dataset described in Section 3.

The two selected networks are trained under four different data scenarios:

S1 Train on synthetic data.

Table G.2: Relevant hyperparameters and the chosen values. For the learning rate and weight decay we try all permutations of the specified values.

Parameter	**Value**
Learning Rate (η)	$[10^{-3}, 10^{-2}, 10^{-1}]$
Momentum	0.9
Weight Decay	$[10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$
Dropout Rate	0.5
Batch Size	32
Epochs	50

S2 Train on real data.

S3 Train on synthetic and real data.

S4 Train on synthetic data, and fine-tune on real data.

The validation and test splits consist of both real and synthetic data for all data scenarios. By testing these different data scenarios we hope to determine the effect of the synthetic data, and how to best utilize the small amount of real life data which may be available.

For each method and scenario we utilize the hyperparameters shown in Table G.2 and perform grid search over the learning rate, η, and the weight decay. For DGCNN we set k to 20, while for PointNet we weight the regularization loss L_{reg} by 0.001. The models are trained for 50 epochs using Stochastic Gradient Descent (SGD) with Momentum, and cosine annealing [54] the learning rate from η to $\eta \cdot 10^{-2}$, and the Cross Entropy loss objective. We handle the class-imbalance between the normal pipes and three defects by weighing the loss objective differently for each class. The class weights are set as the proportion of class samples compared to the class with the most samples. Lastly, the data is augmented during training by jittering each point with noise from a Gaussian distribution, with zero mean and 0.02 standard deviation. For scenario 1, 2, and 3 we select the model which achieved the best validation loss. For scenario 4 we take the best performing model in scenario 1 and fine-tune it, with identical parameters except the selected η, which is multiplied by 10^{-1}.

We evaluate the models by considering their confusion matrices on the real test data as shown in Figure G.7-G.8, as well as the precision, recall, and F1-score in Table G.3-G.4. The metrics are calculated as the average of the binary metrics for each class, where each class is weighted by the proportion of the class in the dataset. We present the resulting metrics for the real test data, as well as for the full test data split. Lastly, we also investigate the effect of the ratio of real life data used when fine-tuning the models in data scenario 4. We investigate using between 0% (*i.e.* no fine-tuning) up to 100% of the real training data, in increments of 10%. The resulting metrics for the real data test split are shown in Figure G.6.

Table G.3: Performance of the PointNet and DGCNN networks on the real data test split, for all four data scenarios. All metrics are the weighted average across all classes.

Model	Precision	Recall	F1
PointNet-S1	3.58	15.88	5.25
DGCNN-S1	29.02	20.62	17.65
PointNet-S2	2.72	16.49	4.67
DGCNN-S2	25.31	50.31	33.68
PointNet-S3	28.61	32.16	30.23
DGCNN-S3	34.55	22.27	16.66
PointNet-S4	23.17	27.42	24.24
DGCNN-S4	39.69	26.19	23.58

Table G.4: Performance of the PointNet and DGCNN networks on the entire data test split, for all four data scenarios. All metrics are the weighted average across all classes.

Model	Precision	Recall	F1
PointNet-S1	8.00	17.21	6.70
DGCNN-S1	57.57	56.73	57.09
PointNet-S2	2.77	16.64	4.75
DGCNN-S2	25.05	50.05	33.39
PointNet-S3	34.36	32.40	31.65
DGCNN-S3	58.72	57.52	58.67
PointNet-S4	28.37	36.11	30.98
DGCNN-S4	50.37	36.61	35.25

6 Discussion

From the results it is evident that the DGCNN network consistently outperforms the PointNet network. Even the best performing case of PointNet, trained using data scenario 3, which scores the highest F1 score, consistently avoids predicting the rubber rings. This is a general theme throughout the trained PointNet networks, which in all other cases stick to predicting one or two classes. Comparatively, the DGCNN networks makes more well rounded predictions, with only DGCNN-S2 consistently predicting a single class. This is reflected by the consistently high metrics. Therefore, it appears that there is a clear benefit of the EdgeConv layers for the sewer defect classification task. This makes sense as both the local and the global structure is affected by defects, due to shadowing of the sensor and changes to the pipe itself.

When looking into the different data strategies, it is found that using either only synthetic or real data is a poor strategy. Instead the best results were obtained by pre-training on synthetic data, followed by fine-tuning on real data. This led to a consistent improvement over both data scenario 1 and 3 on the real data. Looking at Figure G.6, we see that the ratio of real point cloud data used to fine-tune the DGCNN

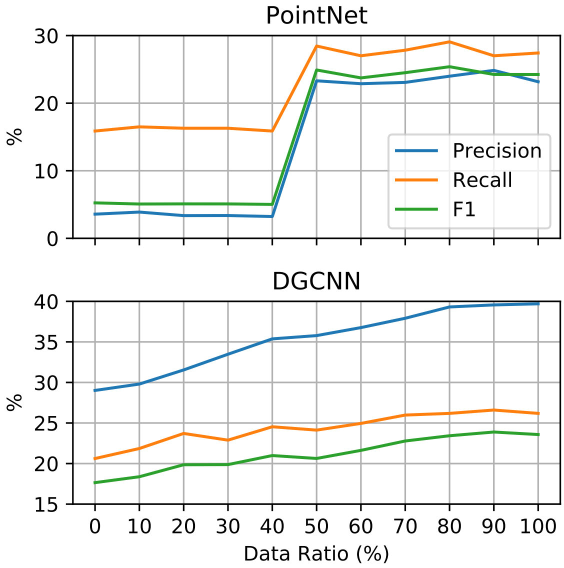


Fig. G.6: Plot of the evaluation metrics, when increasing the ratio of the real training data used to fine-tune the networks.

network is proportional to the metric performance. However, the PointNet network again converges to a point where only one or two classes are predicted, as seen in Figure G.8d. We can therefore conclude that synthetic training data can be used to bootstrap the training process of a 3D sewer defect classifier.

However, the networks are not a perfect classifier, as there are several failure points. As mentioned earlier, only one of the PointNets managed to converge to a usable classifier, with the rest instead simply predicting one or two classes. Conversely, the DGCNN converge to a more usable classifier. However, the best performing model, DGCNN-S4, is biased towards the defect classes, with only 9% correctly predicted non-defective pipes in the real data. This may be due to some defects occurring quite far into the pipe, though still visible to the sensor. In these cases the effect on the recorded point cloud, such as shadowing, will be quite subtle, and more easily confused with a normal sewer pipe.

7 Conclusion

In this work we investigate the possibility of utilizing modern geometric deep learning techniques in order to detect defects in sewer pipes, using a combination of synthetic and real point cloud data. We compare two network architectures, PointNet and DGCNN, on a new publicly available dataset with 17,000 point clouds and four classes. The dataset is structured such that the majority of the training and validation splits consist of synthetic data, with the majority of the point cloud data of real sewer pipes are reserved for the test data. We conduct a grid search for the hyperparameters, and train the chosen networks under four different training data scenarios, in order to investigate the effect of using synthetic and real training data. We find that the DGCNN networks consistently outperforms the PointNet baseline, when investigating the confusion matrices and metrics. We also find that the best performance is achieved using both synthetic and real training data, specifically when using the real data to fine-tune a network trained on synthetic data. The trained classifiers are, however, not perfect, as they tend to favor classifying defects instead of normal pipes. With these findings we show that both geometric deep learning methods and synthetic training data is viable for training sewer defect classifiers, though more work is needed for the classifiers to become more stable.

Acknowledgments

This work is supported by Innovation Fund Denmark [grant number 8055-00015A] and is part of the Automated Sewer Inspection Robot (ASIR) project. The authors declare no conflict of interests.

References

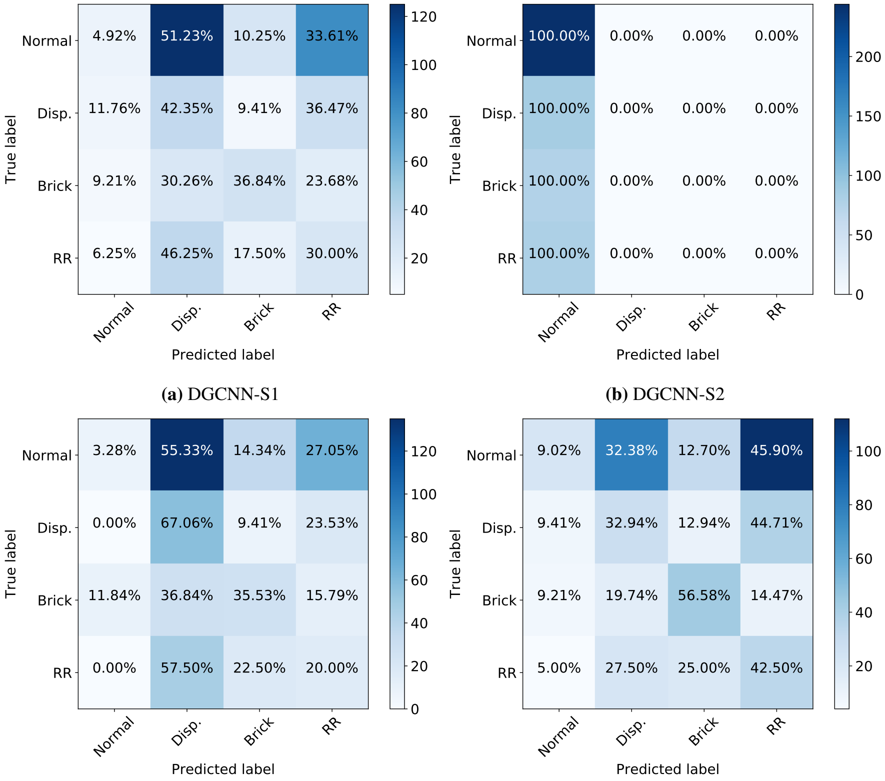
[1] American Society of Civil Engineers, “2017 infrastructure report card - wastewater,” 2017, accessed: 14-11-2020. [Online]. Available: https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf

[2] E. Menendez, J. G. Victores, R. Montero, S. Martínez, and C. Balaguer, “Tunnel structural inspection and assessment using an autonomous robotic system,” *Automation in Construction*, vol. 87, 2018.

[3] X. Qin, G. Wu, J. Lei, F. Fan, X. Ye, and Q. Mei, “A novel method of autonomous inspection for transmission line based on cable inspection robot lidar data,” *Sensors*, vol. 18, no. 2, 2018.

[4] G. G. Garrido, T. Sattar, M. Corsar, R. James, and D. Seghier, “Towards safe inspection of long weld lines on ship hulls using an autonomous robot,” in *21st International Conference on Climbing and Walking Robots*, 2018.

[5] M. Car, L. Markovic, A. Ivanovic, M. Orsag, and S. Bogdan, “Autonomous wind-turbine blade inspection using lidar-equipped unmanned aerial vehicle,” *IEEE Access*, vol. 8, 2020.



(a) DGCNN-S1

(b) DGCNN-S2

(c) DGCNN-S3

(d) DGCNN-S4

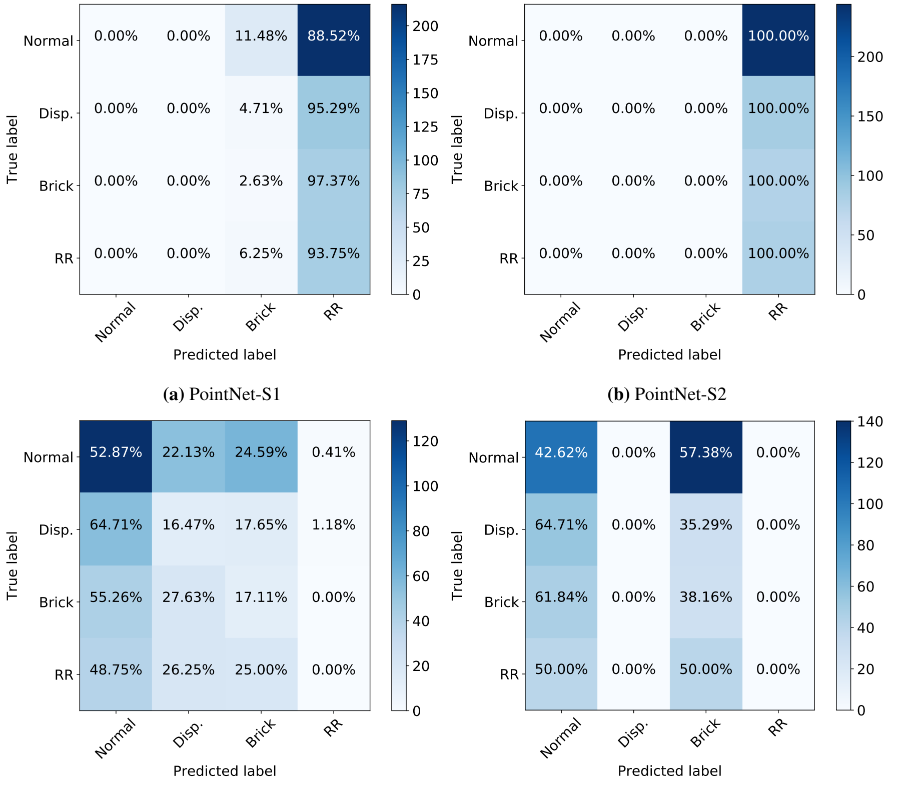
Fig. G.7: Confusion matrices of the real data test split, for the DGCNN architecture and four data scenarios. Disp. and RR denotes Displacement and Rubber Ring. respectively.

[6] M. Nasrollahi, N. Bolourian, and A. Hammad, “Concrete surface defect detection using deep neural network based on lidar scanning,” in *Proceedings of the CSCE Annual Conference, Laval, Greater Montreal, QC, Canada*, 2019.

[7] R. Ravi, D. Bullock, and A. Habib, “Highway and airport runway pavement inspection using mobile lidar,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 43, 2020.

[8] X. Wen, K. Song, M. Niu, Z. Dong, and Y. Yan, “A three-dimensional inspection system for high temperature steel product surface sample height using stereo vision and blue encoded patterns,” *Optik*, vol. 130, 2017.

[9] M. S. Nielsen, I. Nikolov, E. K. Kruse, J. Garnæs, and C. B. Madsen, “High-resolution structure-from-motion for quantitative measurement of leading-edge roughness,” *Energies*, vol. 13, no. 15, 2020.



(a) PointNet-S1

(b) PointNet-S2

(c) PointNet-S3

(d) PointNet-S4

Fig. G.8: Confusion matrices of the real data test split, for the PointNet architecture and four data scenarios. Disp. and RR denotes Displacement and Rubber Ring. respectively.

[10] N. H. Pham, H. M. La, Q. P. Ha, S. N. Dang, A. H. Vo, and Q. H. Dinh, "Visual and 3d mapping for steel bridge inspection using a climbing robot," in *ISARC 2016-33rd International Symposium on Automation and Robotics in Construction*, 2016.

[11] Y. Santur, M. Karaköse, and E. Akın, "Condition monitoring approach using 3d modelling of railway tracks with laser cameras," in *International Conference on Advanced Technology & Sciences (ICAT'16) pp*, 2016.

[12] J. B. Haurum and T. B. Moeslund, "A survey on image-based automation of cctv and sset sewer inspections," *Automation in Construction*, vol. 111, 2020.

[13] K. S. Henriksen, M. S. Lynge, M. D. B. Jeppesen, M. M. J. Allahham, I. A. Nikolov, J. B. Haurum, and T. B. Moeslund, "Generating synthetic point clouds

of sewer networks: An initial investigation," in *Augmented Reality, Virtual Reality, and Computer Graphics*. Cham: Springer International Publishing, 2020.

[14] CamBoard, "Development kit brief camboard pico flexx," 2018, accessed: 25-11-2020. [Online]. Available: https://pmdtec.com/picofamily/wp-content/uploads/2018/03/PMD_DevKit_Brief_CB_pico_flexx_CE_V0218-1.pdf

[15] S. K. Sinha and P. W. Fieguth, "Automated detection of cracks in buried concrete pipe images," *Automation in Construction*, vol. 15, no. 1, 2006.

[16] ——, "Neuro-fuzzy network for the classification of buried pipe defects," *Automation in Construction*, vol. 15, no. 1, 2006.

[17] ——, "Segmentation of buried concrete pipe images," *Automation in Construction*, vol. 15, no. 1, 2006.

[18] T.-C. Su, M.-D. Yang, T.-C. Wu, and J.-Y. Lin, "Morphological segmentation based on edge detection for sewer pipe defects on cctv images," *Expert Systems with Applications*, vol. 38, no. 10, 2011.

[19] M.-D. Yang and T.-C. Su, "Automated diagnosis of sewer pipe defects based on machine learning approaches," *Expert Systems with Applications*, vol. 35, no. 3, 2008.

[20] W. Wu, Z. Liu, and Y. He, "Classification of defects with ensemble methods in the automated visual inspection of sewer pipes," *Pattern Analysis and Applications*, vol. 18, no. 2, Dec. 2013.

[21] J. Myrans, R. Everson, and Z. Kapelan, "Automated detection of faults in sewers using cctv image sequences," *Automation in Construction*, vol. 95, 2018.

[22] S. I. Hassan, L. M. Dang, I. Mehmood, S. Im, C. Choi, J. Kang, Y.-S. Park, and H. Moon, "Underground sewer pipe condition assessment based on convolutional neural networks," *Automation in Construction*, vol. 106, 2019.

[23] D. Li, A. Cong, and S. Guo, "Sewer damage detection from imbalanced cctv inspection data using deep convolutional neural networks with hierarchical classification," *Automation in Construction*, vol. 101, 2019.

[24] S. S. Kumar, M. Wang, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. C. P. Cheng, "Deep learning&#x2013;based automated detection of sewer defects in cctv videos," *Journal of Computing in Civil Engineering*, vol. 34, no. 1, 2020.

[25] M. Wang and J. C. P. Cheng, "A unified convolutional neural network integrated with conditional random field for pipe defect segmentation," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 2, 2020.

[26] O. Duran, K. Althoefer, and L. D. Seneviratne, “Pipe inspection using a laser-based transducer and automated analysis techniques,” *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 3, 2003.

[27] ——, “Automated pipe inspection using ann and laser data fusion,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*, vol. 5, 2004.

[28] ——, “Automated pipe defect detection and categorization using camera/laser-based profiler and artificial neural network,” *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, 2007.

[29] M. Lepot, N. Stanić, and F. H. Clemens, “A technology for sewer pipe inspection (part 2): Experimental assessment of a new laser profiler for sewer defect detection and quantification,” *Automation in Construction*, vol. 73, 2017.

[30] A. D. Tezerjani, M. Mehrandezh, and R. Paranjape, “Defect detection in pipes using a mobile laser-optics technology and digital geometry,” *MATEC Web of Conferences*, vol. 32, 2015.

[31] A. Ahrary, Y. Kawamura, and M. Ishikawa, “A laser scanner for landmark detection with the sewer inspection robot kantaro,” in *2006 IEEE/SMC International Conference on System of Systems Engineering*, 2006.

[32] M. Kolesnik and G. Baratoff, “Online distance recovery for a sewer inspection robot,” in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 1, 2000.

[33] S. Iyer, S. K. Sinha, M. K. Pedrick, and B. R. Tittmann, “Evaluation of ultrasonic inspection and imaging systems for concrete pipes,” *Automation in Construction*, vol. 22, 2012, planning Future Cities-Selected papers from the 2010 eCAADe Conference.

[34] M. S. Khan and R. Patil, “Acoustic characterization of pvc sewer pipes for crack detection using frequency domain analysis,” in *2018 IEEE International Smart Cities Conference (ISC2)*, 2018.

[35] ——, “Statistical analysis of acoustic response of pvc pipes for crack detection,” in *SoutheastCon 2018*, 2018.

[36] D. Alejo, F. Caballero, and L. Merino, “Rgbd-based robot localization in sewer networks,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[37] D. Alejo, G. Mier, C. Marques, F. Caballero, L. Merino, and P. Alvito, *SIAR: A Ground Robot Solution for Semi-autonomous Inspection of Visitable Sewers*. Cham: Springer International Publishing, 2020.

[38] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[40] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017.

[42] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[45] Y. Wang, W. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[46] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, Oct. 2019.

[47] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, 2017.

[48] W. Cao, Z. Yan, Z. He, and Z. He, "A comprehensive survey on geometric deep learning," *IEEE Access*, vol. 8, 2020.

[49] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[50] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, “Structured domain randomization: Bridging the reality gap by context-aware synthetic data,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[51] S. Beery, Y. Liu, D. Morris, J. Piavis, A. Kapoor, M. Meister, N. Joshi, and P. Perona, “Synthetic examples improve generalization for rare classes,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[52] V. Barnett and T. Lewis, *Outliers in Statistical Data*, ser. Wiley Series in Probability and Statistics. Wiley, 1984.

[53] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*. John wiley & sons, 2005, vol. 589.

[54] I. Loshchilov and F. Hutter, “SGDR: stochastic gradient descent with warm restarts,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

SUMMARY

The sewerage infrastructure is one of the critical infrastructures of modern society, which most people rarely consider. However, due to its immense size regular inspection of the sewer pipes is impossible. This thesis focuses on using Computer Vision to automate sewer inspections through two considered modalities: images and point clouds. Computer vision aided sewer inspections has been researched for over three decades but has yet to be widely adopted by professional inspectors.

In this thesis, the fundamental historic trends and hindrances were investigated, covering the algorithmic trends and the lack of public code and datasets as well as no common evaluation protocols. These hindrances were broken down by the release of the first publicly available image and point cloud sewer defect classification datasets, the introduction of domain influenced evaluation metrics, and open-sourcing the developed code. This has consequently made the automated sewer inspection domain far more accessible. Finally, two novel graph-based computer vision algorithms were developed for automating parts of the sewer inspection process leading to significant improvements over prior methods.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-919-6

AALBORG UNIVERSITY PRESS