



Aalborg Universitet

AALBORG UNIVERSITY  
DENMARK

## Water Level Estimation in Sewer Pipes using Deep Convolutional Neural Networks

Haurum, Joakim Bruslund; Bahnsen, Chris Holmberg; Pedersen, Malte; Moeslund, Thomas B.

*Published in:*  
Water

*DOI (link to publication from Publisher):*  
[10.3390/w12123412](https://doi.org/10.3390/w12123412)

*Creative Commons License*  
CC BY 4.0

*Publication date:*  
2020

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

[Link to publication from Aalborg University](#)

*Citation for published version (APA):*  
Haurum, J. B., Bahnsen, C. H., Pedersen, M., & Moeslund, T. B. (2020). Water Level Estimation in Sewer Pipes using Deep Convolutional Neural Networks. *Water*, 12(12), Article 3412. <https://doi.org/10.3390/w12123412>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

### Take down policy

If you believe that this document breaches copyright please contact us at [vbn@aub.aau.dk](mailto:vbn@aub.aau.dk) providing details, and we will remove access to the work immediately and investigate your claim.

Article

# Water Level Estimation in Sewer Pipes Using Deep Convolutional Neural Networks

Joakim Bruslund Haurum <sup>\*ID</sup>, Chris H. Bahnsen <sup>ID</sup>, Malte Pedersen <sup>ID</sup>  
and Thomas B. Moeslund <sup>ID</sup>

Visual Analysis of People (VAP) Lab, Aalborg University, Rendsburggade 14, 9000 Aalborg, Denmark; cb@create.aau.dk (C.H.B.); mape@create.aau.dk (M.P.); tbm@create.aau.dk (T.B.M.)

\* Correspondence: joha@create.aau.dk

Received: 16 October 2020; Accepted: 30 November 2020; Published: 4 December 2020



**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.

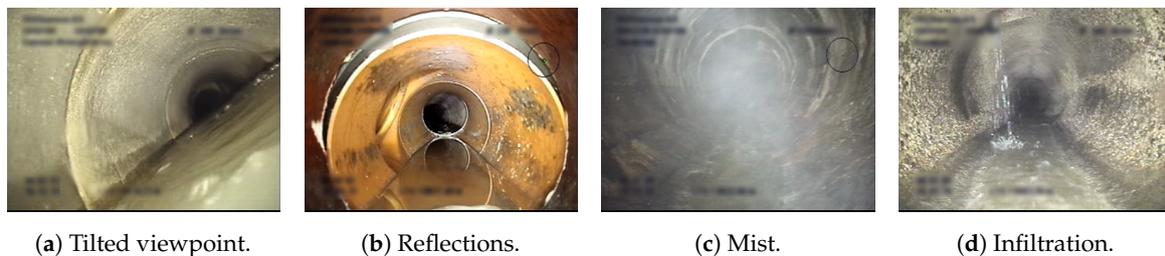
**Keywords:** sewer pipes; convolutional neural networks; random forests; water level; sewer inspection standards

## 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 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 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.



**Figure 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.

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 1,440 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 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>).

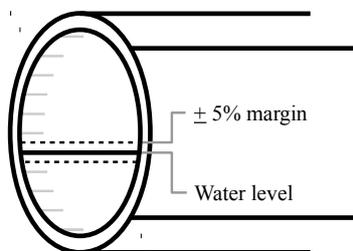
The remainder of the article is structured as follows. In Section 2, we introduce our dataset. In Section 3, we describe the proposed methods, loss functions, and the training procedure. In Section 4, we detail the evaluation metrics and present the experimental results. In Section 5, we analyze and 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 2.



**Figure 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 1. The data in the training, validation, and test splits all come from unique sewer pipes and inspections.

**Table 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

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 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.

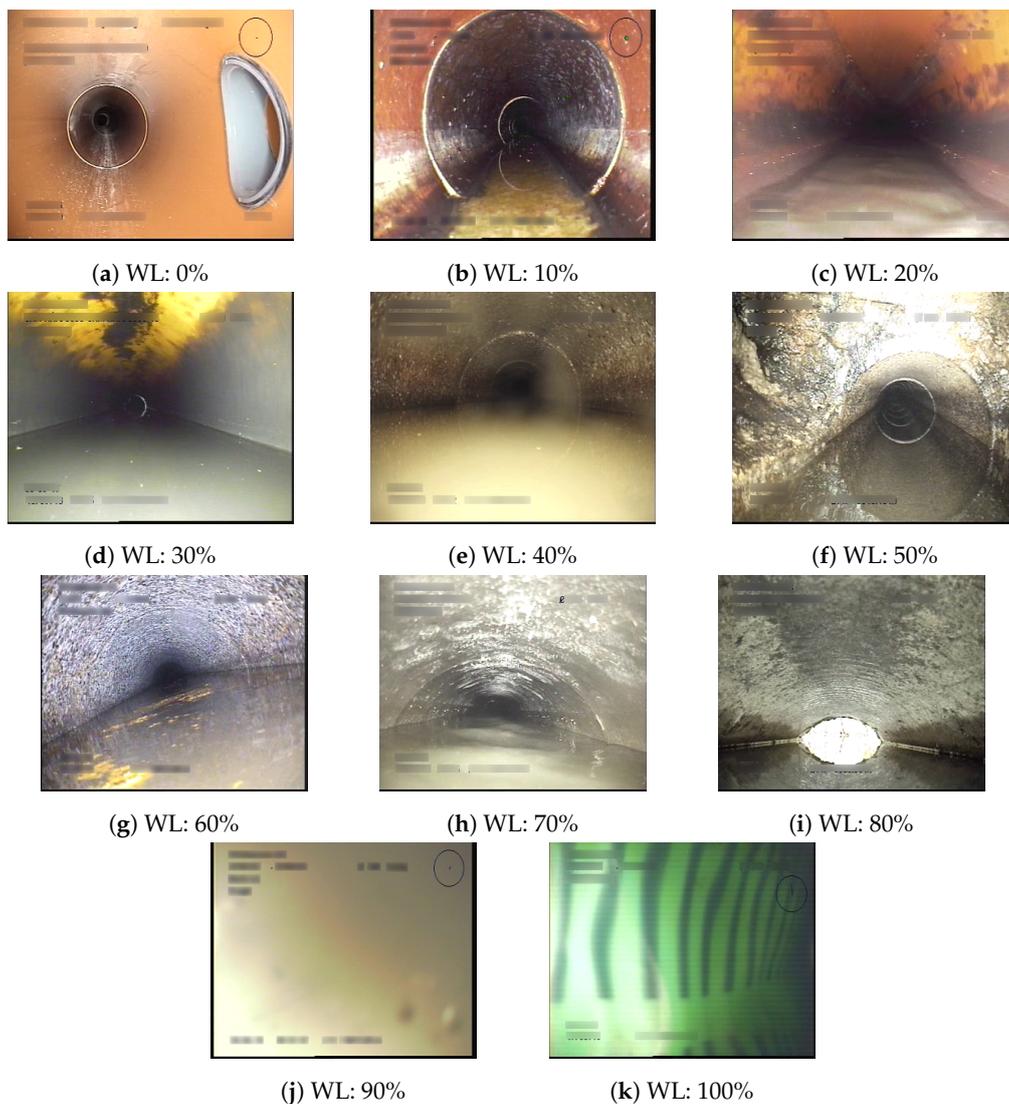
It is possible to make a near perfect one-to-one mapping between the two standards with the exception being the 2015 class with  $\geq 30\%$  water level. As the 2010 annotations have a  $\pm 5\%$  point uncertainty margin, the 30% class may contain data from water levels as low as 26%. We choose to accept the risk of this and acknowledge it as a source of extra noise. With this mapping, the dataset presented in Table 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 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.

**Table 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

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 water in the sewer pipe and not by reading the textual metadata. A range of examples from the dataset are shown in Figure 3.



**Figure 3.** Images from the dataset that show the inter-class variation. WL is the annotated water level.

### 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 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

$$f_{CE} = - \sum_{c=1}^C w_c y_c \log(p_c) \quad (1)$$

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  $\ell_1$  and  $\ell_2$  loss functions, respectively. MAE and MSE are defined as

$$f_{\text{MAE}}(x) = |x| \quad (2)$$

$$f_{\text{MSE}}(x) = x^2 \quad (3)$$

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 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 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.

**Table 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

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 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  $4 \times 4$  grid with filters using 4 scales and 8 orientations. The input image is downscaled to  $128 \times 128$  pixels and converted to grayscale as described in [28], which results in a 512 dimensional GIST feature vector.

**Table 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	$[\sqrt{d}, \log_2(d), d/3, d]$

All classification models are trained by minimizing a weighted CE loss where the class weights are calculated as

$$w_c = \frac{\max(N_i)}{N_c}, i \forall [1, 2, \dots, C], \quad (4)$$

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 tree-based models are found from the model with the lowest validation loss among the models in the aforementioned hyperparameter search, see Table 5.

**Table 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	$\sqrt{d}$	$d$	$\sqrt{d}$	$d$	$d$	$d$

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 6–9. The results of the CNNs trained from scratch can be found in Appendix 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 (5)–(7). TP, FP, and FN denote the true positive, false positive, and false negative predictions, respectively.

$$P = \frac{TP}{TP + FP} \quad (5)$$

$$R = \frac{TP}{TP + FN} \quad (6)$$

$$F1 = \frac{2 P R}{P + R} \quad (7)$$

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 (8)–(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.

$$\text{micro-P} = \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C TP_c + FP_c} \tag{8}$$

$$\text{micro-R} = \frac{\sum_{c=1}^C TP_c}{\sum_{c=1}^C TP_c + FN_c} \tag{9}$$

$$\text{micro-F1} = \frac{2 \text{ micro-P micro-R}}{\text{micro-P} + \text{micro-R}} \tag{10}$$

The Macro-F1 metric is calculated as the arithmetic mean of the per-class F1-metrics as shown in Equation (11). This results in an equal weighting for each class, thereby causing the Macro-F1 metric to be more sensitive to the rare classes.

$$\text{Macro-F1} = \frac{1}{C} \sum_{c=1}^C F1_c \tag{11}$$

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  $\pm 5\%$  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 6 and the F1 score for each class under the different training variations are shown in Tables 7–9 and Tables A1–A4. 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 4.

Table 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	<b>37.41</b>	<b>30.61</b>	<b>30.00</b>	73.03	60.93
ResNet34-FT	37.37	35.54	28.69	28.00	76.36	61.88
ResNet50-FT	<b>39.70</b>	36.50	27.07	26.27	<b>79.29</b>	<b>62.88</b>

Table 7. Per-class F1 score for each method—Class10 training setting.

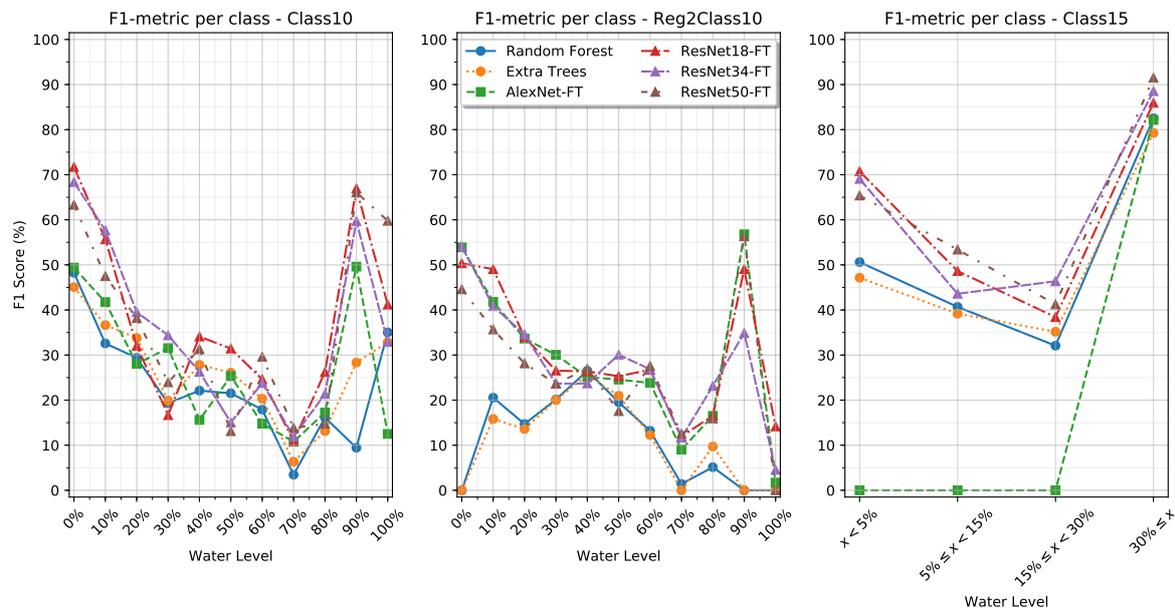
Method \ Water Level	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Random Forest	48.25	32.58	29.38	19.32	22.11	21.52	17.89	3.45	16.13	9.43	35.02
Extra Trees	45.05	36.65	33.79	19.88	27.84	26.09	20.32	6.35	13.14	28.33	32.85
AlexNet-FT	49.38	41.75	28.05	31.54	15.60	25.32	14.77	10.77	17.27	49.61	12.50
ResNet18-FT	<b>71.72</b>	55.67	32.05	16.67	<b>34.10</b>	<b>31.43</b>	24.62	10.89	<b>26.23</b>	<b>66.94</b>	41.18
ResNet34-FT	68.38	<b>57.65</b>	<b>39.52</b>	<b>34.38</b>	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	<b>29.63</b>	<b>13.86</b>	14.74	66.08	<b>59.77</b>

**Table 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	<b>30.05</b>	25.11	24.56	23.81	9.02	16.51	<b>56.79</b>	1.69
ResNet18-FT	50.35	<b>49.04</b>	34.00	26.54	26.36	25.31	26.73	11.94	16.55	49.06	<b>14.16</b>
ResNet34-FT	<b>54.09</b>	40.84	<b>34.55</b>	23.66	23.70	<b>30.08</b>	26.84	11.68	<b>23.13</b>	34.90	4.55
ResNet50-FT	44.59	35.68	28.14	23.65	<b>26.96</b>	17.56	<b>27.59</b>	<b>12.58</b>	15.91	56.32	0.00

**Table 9.** Per-class F1 score for each method—Class15 training setting.

Method\Water Level	WL < 5%	5% ≤ WL < 15%	15% ≤ WL < 30%	30% ≤ 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	<b>70.75</b>	48.62	38.41	85.94
ResNet34-FT	69.06	43.59	<b>46.36</b>	88.53
ResNet50-FT	65.38	<b>53.39</b>	41.24	<b>91.51</b>



**Figure 4.** Per-class F1-metric for all methods under the different training settings.

### 5. Discussion

From the results presented in Table 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 8 and 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 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.H.B., 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.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Results for CNNs Trained From Scratch

**Table A1.** 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	<b>54.41</b>
ResNet34-S	<b>29.90</b>	25.72	<b>20.71</b>	<b>19.80</b>	<b>72.02</b>	53.35
ResNet50-S	29.29	<b>26.20</b>	19.79	18.92	68.18	48.31

**Table A2.** 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	<b>55.45</b>	<b>41.67</b>	18.18	8.20	<b>27.07</b>	20.83	8.00	<b>16.04</b>	10.53	16.20	37.31
ResNet34-S	51.68	38.64	<b>30.59</b>	<b>20.00</b>	16.54	<b>24.18</b>	11.76	11.49	8.85	22.62	<b>46.62</b>
ResNet50-S	43.22	32.00	28.24	10.37	24.51	14.93	<b>17.89</b>	6.11	<b>13.89</b>	<b>56.00</b>	40.99

**Table A3.** 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	<b>33.33</b>	<b>37.56</b>	18.10	20.87	<b>17.92</b>	<b>23.04</b>	16.43	11.01	11.37	3.97	<b>2.04</b>
ResNet34-S	20.69	36.84	<b>28.18</b>	20.72	12.67	21.05	13.98	<b>16.42</b>	<b>12.97</b>	32.22	<b>2.04</b>
ResNet50-S	18.64	26.83	19.81	<b>23.85</b>	16.55	22.86	<b>17.02</b>	7.75	2.68	<b>52.11</b>	0.00

**Table A4.** Per-class F1 score for the CNNs trained from scratch—Class15 training setting.

Method \ Water Level	WL < 5%	5% ≤ WL < 15%	15% ≤ WL < 30%	30% ≤ WL
AlexNet-S	0.00	0.00	0.00	82.14
ResNet18-S	<b>53.08</b>	<b>45.57</b>	33.49	85.49
ResNet34-S	52.78	38.92	<b>34.78</b>	<b>86.92</b>
ResNet50-S	50.49	25.00	33.90	83.87

## References

1. Statistisches Bundesamt. *Öffentliche Wasserversorgung und Öffentliche Abwasserentsorgung-Strukturdaten zur Wasserwirtschaft 2016*; Technical Report; Fachserie 19 Reihe 2.1.3.; Statistisches Bundesamt: Wiesbaden, Germany, 2018.
2. American Society of Civil Engineers (ASCE). *2017 Infrastructure Report Card-Wastewater*; ASCE: Reston, VA, USA, 2017. Available online: <https://www.infrastructurereportcard.org/wp-content/uploads/2017/01/Wastewater-Final.pdf> (accessed on 27 March 2020).
3. United States Environmental Protection Agency (EPA). *Fact Sheet: Asset Management for Sewer Collection Systems*; EPA: Washington, DC, USA, 2002.
4. Haurum, J.B.; Moeslund, T.B. A Survey on Image-Based Automation of CCTV and SSET Sewer Inspections. *Autom. Constr.* **2020**, *111*, 103061. [CrossRef]

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): Brussels, Belgium, 2003.
6. Kirstein, S.; Müller, K.; Walecki-Mingers, M.; Deserno, T.M. Robust adaptive flow line detection in sewer pipes. *Autom. Constr.* **2012**, *21*, 24–31. [[CrossRef](#)]
7. Halfawy, M.R.; Hengmechai, J. Integrated Vision-Based System for Automated Defect Detection in Sewer Closed Circuit Television Inspection Videos. *J. Comput. Civ. Eng.* **2015**, *29*. [[CrossRef](#)]
8. Li, D.; Cong, A.; Guo, S. Sewer damage detection from imbalanced CCTV inspection data using deep convolutional neural networks with hierarchical classification. *Autom. Constr.* **2019**, *101*, 199–208. [[CrossRef](#)]
9. Xie, Q.; Li, D.; Xu, J.; Yu, Z.; Wang, J. Automatic detection and classification of sewer defects via hierarchical deep learning. *IEEE Trans. Autom. Sci. Eng.* **2019**, *16*, 1836–1847. [[CrossRef](#)]
10. Ji, H.; Yoo, S.; Lee, B.J.; Koo, D.; Kang, J.H. Measurement of Wastewater Discharge in Sewer Pipes Using Image Analysis. *Water* **2020**, *12*, 1771. [[CrossRef](#)]
11. Nguyen, L.S.; Schaeli, B.; Sage, D.; Kayal, S.; Jeanbourquin, D.; Barry, D.A.; Rossi, L. Vision-based system for the control and measurement of wastewater flow rate in sewer systems. *Water Sci. Technol.* **2009**, *60*, 2281–2289. [[CrossRef](#)] [[PubMed](#)]
12. Jeanbourquin, D.; Sage, D.; Nguyen, L.; Schaeli, B.; Kayal, S.; Barry, D.A.; Rossi, L. Flow measurements in sewers based on image analysis: Automatic flow velocity algorithm. *Water Sci. Technol.* **2011**, *64*, 1108–1114. [[CrossRef](#)]
13. Lin, Y.T.; Lin, Y.C.; Han, J.Y. Automatic water-level detection using single-camera images with varied poses. *Measurement* **2018**, *127*, 167–174. [[CrossRef](#)]
14. Gilmore, T.E.; Birgand, F.; Chapman, K.W. Source and magnitude of error in an inexpensive image-based water level measurement system. *J. Hydrol.* **2013**, *496*, 178–186. [[CrossRef](#)]
15. Bruinink, M.; Chandarr, A.; Rudinac, M.; van Overloop, P.; Jonker, P. Portable, automatic water level estimation using mobile phone cameras. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015; pp. 426–429. [[CrossRef](#)]
16. Sirazitdinova, E.; Pesic, I.; Schwehn, P.; Song, H.; Satzger, M.; Sattler, M.; Weingärtner, D.; Deserno, T.M. Sewer Discharge Estimation by Stereoscopic Imaging and Synchronized Frame Processing. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 602–613. [[CrossRef](#)]
17. Khorchani, M.; Blanpain, O. Free surface measurement of flow over side weirs using the video monitoring concept. *Flow Meas. Instrum.* **2004**, *15*, 111–117. [[CrossRef](#)]
18. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [[CrossRef](#)]
19. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755. [[CrossRef](#)]
20. Dansk Vand og Spildevandsforening (DANVA). *Fotomanualen: TV-Inspektion af Afløbsledninger*, 6th ed.; Dansk Vand og Spildevandsforening (DANVA): Skanderborg, Denmark, 2010.
21. Dansk Vand og Spildevandsforening (DANVA). *Fotomanualen: TV-Inspektion af Afløbsledninger*, 7th ed.; Dansk Vand og Spildevandsforening (DANVA): Skanderborg, Denmark, 2015.
22. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Dutchess County, NY, USA, 2012; pp. 1097–1105.
23. He, K.; Sun, J. Convolutional Neural Networks at Constrained Time Cost. *arXiv* **2014**, arXiv:cs.CV/1412.1710.
24. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
25. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
26. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely randomized trees. *Mach. Learn.* **2006**, *63*, 3–42. [[CrossRef](#)]
27. Oliva, A.; Torralba, A. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [[CrossRef](#)]

28. Myrans, J.; Everson, R.; Kapelan, Z. Automated detection of faults in sewers using CCTV image sequences. *Autom. Constr.* **2018**, *95*, 64–71. [[CrossRef](#)]
29. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2015**, arXiv:1412.6980.
30. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H.; Larochelle, H.; Beygelzimer, A., d'Alche-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
31. Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; et al. API design for machine learning software: Experiences from the scikit-learn project. ECML PKDD Workshop: Languages for Data Mining and Machine Learning. *arXiv* **2013**, arXiv:1309.0238
32. Tsuchiya, Y. Lear-Gist-Python. Available online: <https://github.com/tuttiieee/lear-gist-python> (accessed on 2 December 2020).
33. van der Steen, A.J.; Dirksen, J.; Clemens, F.H. Visual sewer inspection: Detail of coding system versus data quality? *Struct. Infrastruct. Eng.* **2014**, *10*, 1385–1393. [[CrossRef](#)]

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).