Generic Object Detection and Segmentation for Real-World Environments

Johansen, Anders Skaarup

DOI (link to publication from Publisher):
10.54337/aau561828875

Publication date:
2023

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

GENERIC OBJECT DETECTION AND SEGMENTATION FOR REAL-WORLD ENVIRONMENTS

BY
ANDERS SKAARUP JOHANSEN

DISSERTATION SUBMITTED 2023

AALBORG UNIVERSITY
DENMARK

Generic Object Detection and Segmentation for Real-World Environments

Ph.D. Dissertation
Anders Skaarup Johansen

Dissertation submitted August 13, 2023

Dissertation submitted: August 13, 2023

PhD supervisor:: Prof. Kamal Nasrollahi
Aalborg University

PhD committee: Professor Georgios Triantafyllidis (chair)
Aalborg University, Denmark

Professor Jana Kosecká
George Mason University, USA

Professor Serge Belongie
University of Copenhagen, Denmark

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Architecture, Design and Media Technology

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-657-7

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, 2023

Curriculum Vitae

Anders Skaarup Johansen

Anders Skaarup Johansen received his bachelor's in Medialogy in 2017 and finished his master's degree in Medialogy in 2019, both from Aalborg University in Denmark. While studying for his master's degree he spent a semester as an intern at HSA-Systems in Aalborg, where he assisted in the development of super-resolution algorithms for deployment on edge devices. His master's thesis focused on trust estimation for collaborative robots working in close proximity to people. After completing his master's he was employed as a research assistant at the Visual Analysis and Perception (VAP) laboratory at the Department of Architecture, Design, and Media Technology at Aalborg University from 2019-2020, where he worked on designing a sensor package for autonomous sewer inspection robots. Prior to becoming a PhD-student, Anders' research primarily focused on using computer-vision systems for scene understanding and automation of the inspection process. During his time at VAP-Lab Anders has supervised and co-supervised more than 15 undergraduate and graduate projects, primarily focused on applications of segmentation/detection for human-machine interfaces and inspection. The PhD was done at VAP-Lab, and during his PhD studies, Anders collaborated with the Research Department at Milestone Systems A/S, Brøndby, Denmark, and the Computer Vision Center at the University of Barcelona, Barcelona, Spain. In addition to his research, he organized and ran a public competition at the "Real-World Surveillance: Challenges and Applications"-Workshop at ECCV 2020 as a part of ChaLearn: Looking at People. His main interests revolve around computer vision and robotics, in particular helping automated systems understand the world they inhabit.

Curriculum Vitae

Abstract

The advances in the field of computer vision have shown great potential for solving complex problems across a wide range of tasks, and the application of such techniques could greatly benefit society at large. This thesis focuses on research in the field of computer vision, particularly addressing the challenges of computer vision systems in real-world contexts. The research contributions center around three key topics, namely: joint segmentation and super-resolution, transformer-based models in the video domain, and the impact of thermal concept drift on the performance of object-detection algorithms. A key objective is to advance the performance and adaptability of vision systems for real-world application. This work presents several notable contributions. Firstly, novel frameworks like the Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR) are introduced, enhancing segmentation through joint optimization and achieving State-of-the-Art accuracy on challenging datasets like Cityscapes and IDD-Lite. The Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR) framework enhances real-world super-resolution by incorporating semantic guidance, resulting in improved perceptual quality and reduced noise, achieving State-of-the-Art results on real-world images. A comprehensive survey of Video Transformer (VT) methods is conducted, investigating their potential and limitations in video analysis, with an emphasis on addressing computational challenges and high-dimensional redundancy inherent in video data. Additionally, the study of thermal concept drift introduces the largest thermal dataset for Long-Term Drift (LTD) analysis, shedding light on weather-related drift factors and their impact on various vision tasks. A concept drift challenge is also organized, enabling detailed analysis of object detection under thermal concept drift, considering key weather conditions and object configurations. Lastly, the exploration of weather-aware conditioning methods aims to enhance object detection under thermal concept drift, revealing challenges in effectively modeling weather-aware representations through auxiliary weather prediction.

In conclusion, this thesis makes substantial contributions to the field of computer vision, advancing our capabilities in semantic segmentation, super-resolution, transformer-based models for video, and analysis of thermal concept drift. The research not only introduces innovative frameworks but also provides comprehensive datasets and insightful analyses that collectively enrich our understanding and pave the way for more robust and adaptable visual analysis for real-world applications.

Resumé

Fremskridtene inden for computer vision-feltet har vist stort potentiale for at løse komplekse problemer på tværs af en bred vifte af opgaver, og anvendelsen af sådanne teknikker kunne i høj grad gavne samfundet som helhed. Denne afhandling fokuserer på forskning inden for computer vision-feltet og adresserer især udfordringerne ved computer vision-systemer i virkelighedsnære kontekster. Forskningsbidragene drejer sig om tre centrale emner, nemlig: Fælles optimering segmentering og superopløsning, transformer-baserede modeller i videodomænet og påvirkningen af termisk konceptdrift på ydeevnen af objekt-detektions-algoritmer. Et centralt mål er at forbedre ydeevnen og tilpasningsevnen af visionsystemer til andvendelse i virkelighedsnære kontekster. Denne afhandling præsenterer flere bemærkelsesværdige bidrag. For det første introduceres nye teknikker som Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR), der forbedrer segmenteringen gennem fælles optimering af segmentering og superopløsning, og opnår State-of-the-Art nøjagtighed på de udfordrende datasæt Cityscapes og IDD-Lite. Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR) forbedrer virkelighedsnær superopløsning ved at inkorporere semantisk vejledning, hvilket resulterer i forbedret billedkvalitet og reduceret støj, og opnår State-of-the-Art resultater på virkelige billeder. En omfattende undersøgelse af Video Transformer (VT) metoder gennemføres, hvor der fokuseres på deres potentiale og begrænsninger inden for videoanalyse, med vægt på at tackle beregningsmæssige udfordringer og redundanse, som er en almen del af videodata. Derudover introducerer vi det største termiske datasæt til analyse af Langtids Termisk Drift (LTD), der belyser vejrrelaterede driftsfaktorer og deres effekt på forskellige computer vision relaterede opgaver. En konkurence inden for konceptdrift organiseres også, der muliggør detaljeret analyse af objektdetektion under termisk konceptdrift og tager hensyn til relevante vejrforhold og objektkonfigurationer. Til sidst undersøges vejrafhængige konditioneringsmetoder for at forbedre objektdetektion under termisk konceptdrift, hvilket afslører udfordringerne der opstår ved modellering af vejrafhængige repræsentationer gennem ekstern vejrprædiktion.

I alt sin helhed bidrager denne afhandling betydeligt til computer vision-feltet ved at udvide vores evner inden for semantisk segmentering, superopløsning, transformer-baserede modeller til video og analyse af termisk koncept-drift. Forskningen introducerer ikke kun innovative teknikker, men leverer også omfattende datasæt og indsigtfulde analyser, der samlet set beriger vores forståelse og baner vejen for mere robust og tilpasningsdygtig visuel analyse til anvendelse i virkelighedsnære kontekster.

Contents

Thesis Details

Thesis Title:	Generic Object Detection and Segmentation for Real-World Environments
Ph.D. Student:	Anders Skaarup Johansen
Supervisors:	Prof. Kamal Nasrollahi, Aalborg University

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

[A] Andreas Aakerberg, **Anders S. Johansen**, Kamal Nasrollahi and Thomas B. Moeslund, "Single-Loss Multi-Task Learning For Improving Semantic Segmentation Using Super-Resolution," *Computer Analysis of Images and Patterns - 19th International Conference (CAIP), Lecture Notes in Computer Science*, vol. 13053, pp. 403–411, 2021.

[B] Andreas Aakerberg, **Anders S. Johansen**, Kamal Nasrollahi, Thomas B. Moeslund, "Semantic Segmentation Guided Real-World Super-Resolution," *IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACV - Workshops*, pp. 449–458, 2022.

[C] Javier Selva, **Anders S. Johansen**, Sergio Escalera, Kamal Nasrollahi, Thomas B. Moeslund and Albert Clapés, "Video Transformers: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, pp. 1–20, 2023.

[D] Ivan Nikolov, Mark P. Philipsen, Jinsong Liu, Jacob V. Dueholm, **Anders S. Johansen**, Kamal Nasrollahi, Thomas B. Moeslund, "Seaons in Drift: A Long-Term Thermal Imaging Dataset for Studying Concept Drift," *Thirty-fifth Conference on Neural Information Processing Systems*, pp. 1–20, 2021.

[E] **Anders S. Johansen**, Julio C. S. Jacques Junior Sergio Escalera, Kamal Nasrollahi, Thomas B. Moeslund, "ChaLearn LAP Seasons in Drift Challenge: Dataset, Design and Results," *European Conference on Computer Vision : Workshop on Real-World Surveillance - Tel Aviv, Israel*, vol. 13805, pp. 755-769, 2022.

[F] **Anders S. Johansen**, Sergio Escalera, Kamal Nasrollahi, "Who cares about the weather?: Inferring Weather Conditions for Weather-Aware Object Detection in Thermal images," *MDPI Applied Sciences: New Trends in Image Processing III (Under Review)*, 2023.

In addition to the main papers, the PhD student has co-authored the following publications which are not a part of the thesis.

[1] Jonathan Eichild Schmidt, Oscar Edvard Mäkinen, Simon Gørtz Flou Nielsen, **Anders Skaarup Johansen**, Kamal Nasrollahi, Thomas B. Moeslund, "Exploring loss functions for optimising the accuracy of Siamese Neural Networks in Re-Identification applications," *SPIE Fourteenth International Conference on Machine Vision (ICMV)*, 2021.

[2] Chris Holmberg Bahnsen, **Anders Skaarup Johansen**, Mark Philip Philipsen, Jesper Wædeled Henriksen, Kamal Nasrollahi, Thomas B. Moeslund, "3D Sensors for Sewer Inspection: A Quantitative Review and Analysis," *Sensors*, vol. 21, no. 7, 2021.

[3] Jesper Wædeled Henriksen, **Anders Skaarup Johansen**, Matthias Rehm, "Pilot Study for Dynamic Trust Estimation in Human-Robot Collaboration," *ACM/IEEE International Conference on Human-Robot Interaction*, no. 15, pp. 242–244, 2020.

[4] **Anders Skaarup Johansen**, Jesper Wædeled Henriksen, Mohammad Ahsanul Haque, Mohammad Naser Sabet Jahromi, Kamal Nasrollahi, Thomas B. Moeslund, "Multimodal Heartbeat Rate Estimation from the Fusion of Facial RGB and Thermal Videos," *SPIE The 11th International Conference on Machine Vision (ICMV)*, vol. 11041, 2019.

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

This dissertation is submitted as a collection of papers in partial fulfillment of a PhD study at the Section of Media Technology, Aalborg University, Denmark. The PhD study was conducted in the period of 2020-2023, mainly in the Visual Analysis and Perception (VAP) Lab at Aalborg University, Aalborg, Denmark. The study also included collaborations with the Research Department at Milestone Systems A/S, Brøndby, Denmark and The Computer Vision Center, at the University of Barcelona, Barcelona, Spain. This PhD fellowship was funded with support from Milestone Systems A/S as a part of the Milestone Research Programme at Aalborg University.

This dissertation covers research in the area of real world applications of object detection and segmentation. Furthermore the work conducted as a part of this dissertation was in part motivated by trying to understand challenges that arise when computer-vision algorithms are applied outside of controlled environments. Notably the dissertation focuses on three main topics: *Joint Learning of Super-Resolution and Semantic Segmentation*, *Transformers in the Video Domain*, and *Concept Drift During Long-Term Deployment of Vision-Systems*.

The PhD period coincided with a tumultuous time in my personal life, where i often doubted the path ahead of me. Consequently, this thesis is in big parts thanks to several individuals whom without, I would not have completed the thesis. A special thanks go to Thomas B. Moeslund who always provided an open door and a comfy chair to discuss anything. Thomas has truly managed to create a unique work environment that truly helped me grow not only as a researcher but also on a personal level. This extends to PhD supervisor prof. Kamal Nasrollahi, who has provided guidance since i was an undergraduate student. Our collaboration has provided opportunities to truly develop my skills as a researcher and communicator. Furthermore i would like to thank Sergio Escalera for the long and fruitful collaboration, as well as the insightful discussions and excellent supervision throughout our collaboration. Another heartfelt thanks goes out to Chris Holmberg Bahnsen, who opened my eyes to the intriguing world of computer vision.

I would also like to express my unending gratitude to my colleagues at VAP-Lab, in particular Andreas Aakerberg and Malte Pedersen who has been like brothers to me, the daily banter and discussions was always interesting and eye-opening. I could always rely on you for advice and presence of mind, even when i manage to lose my travel documents half-way through traveling to the other side of the globe. And Neelu Madan, who always had invaluable insights and feedback when i needed it.

Moreover, I would like to give my heartfelt thanks to my friends Albert Møller, your patience and kindness regardless of circumstances inspires me to this day, Alex Niekrenz, I could always count on you to lend an ear and support me through tough times, and Mikkel Enoch, you have always inspired me to see the bigger picture and aspire to make an impact.

Last but not least, i would like to give a thanks to my loving family who have always supported my decisions and encouraged me to step out of my comfort zone and pursue my dreams.

Dedicated to my family.
Whose unwavering faith and support have been an endless source of strength.

Jens Frandsen Johansen
Louise Skaarup Johansen
Marianne Skaarup Johansen

Anders Skaarup Johansen
Aalborg University, August 13, 2023

Part I

Overview of work

Chapter 1

Introduction

Perception of the environment we inhabit is a crucial skill we rely on in our daily life. Through our senses, we absorb immense volumes of information at any given moment, which our brains seamlessly process and act upon. Even when information may be lacking, we are often able to draw on historical and contextual experiences to make informed judgments, predictions, or decisions. While we take advantage of all of our senses to understand the world around us, our vision plays a crucial role in everyday tasks, such as driving a car, watching a movie, or cooking a meal. For many of us, it is hard to comprehend what the world would be like if we could not visually perceive it. As such, it is natural that vision has become an important component when assessing the quality and efficiency of our work and solutions. To this day, visual inspection conducted manually by human observers remains a common way to perform quality control in the industry or for surveillance applications. However, this is an expensive and non-scalable solution, it typically involves tedious and redundant work, and the outcome is subjective and reliant on the condition of the observer.

In the past decades, cameras started to be installed in many different settings, enabling an individual to monitor multiple streams of data thus optimizing the inspection process. However, this approach remained subjective and costly while leading to the accumulation of a large amount of video and image data. In essence, video streams provide a continuous flow of information which by itself does not provide any inherent understanding. To perceive and comprehend visual information, a vision system must be capable of identifying patterns, textures, and connections, to construct a detailed and nuanced comprehension of the context and content within a scene. Teaching computers how to perceive and understand visual data has been a long-standing topic in computer science, called computer vision. Initially, computer vision systems relied on humans to manually design the means by which the vision system

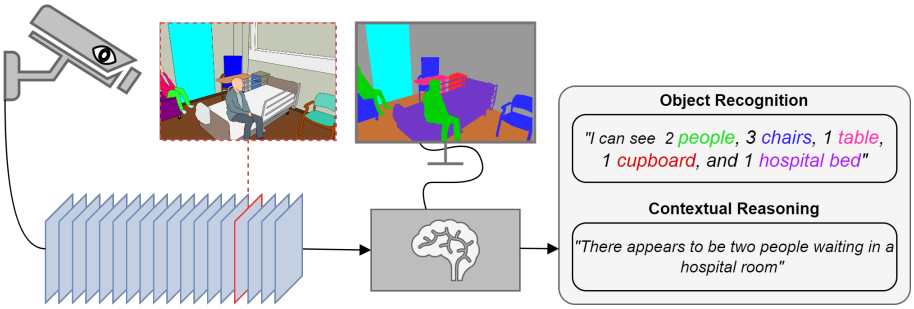


Fig. 1.1: A schematic overview of the computer vision pipeline from start to end. Depicting image capture, the resulting stream of information, pattern recognition, and finally object recognition and contextual understanding.

could extract information. However, with the advent of Machine Learning (ML), we have been able to design intelligent vision systems which can adjust their internal parameters and learn directly from data instead of relying on manually designed components. Modeled after our understanding of the human brain, Artificial Neural Networks (ANNs) allowed us to develop complex vision systems capable of fine-grained understanding and reasoning of visual data. ANNs have surpassed manually designed systems in terms of accuracy and in some cases even outperformed human accuracy [12, 21]. With the combination of cameras and ANNs, we have effectively created artificial eyes to observe the world and artificial brains to understand it.

One of the fundamental concepts in the field of computer vision is object recognition, which enables machines to comprehend and interact with the real world. At its core, object recognition involves the identification and understanding of objects. Two pivotal tasks within the realm of object recognition are object detection and segmentation. Object detection focuses on locating instances of specific objects within an image and drawing bounding boxes around them. This task is crucial for scenarios where precise localization of objects is necessary, such as in autonomous driving, where detecting pedestrians, vehicles, and traffic signs is vital [4, 18]. Segmentation expands this task, by not only identifying objects but also outlining their exact boundaries at a pixel level. Thus allowing a more detailed understanding of object shapes and spatial relationships. This level of granularity is essential in applications such as medical imaging, where segmenting organs or anomalies aids in diagnosis [5, 19]. Object detection, and by extension segmentation, stands as a cornerstone in the field of computer vision, exemplifying the convergence of human-like perception. Applying these methods allow us to extract detailed understanding from visual data streams, lightening the burden of manual human inspections and augmenting current inspection and analysis pipelines.

However, the transition from research to real-world deployment is far from trivial, as the dynamic nature of an unconstrained environment poses several unique challenges and limitations.

1 Real-World Applications of Computer Vision

The performance of computer vision systems is often evaluated on well-established datasets to facilitate broad and detailed comparisons between methods. Large-scale datasets such as ImageNet Large-Scale Visual Recognition Challenge (ImageNet) [21], MicroSoft: Common Objects in Context (MS COCO) [14], and Google Open Images (OpenImages) [13] contain thousands of images captured with different cameras, under different conditions and are as such considered a solid baseline benchmark for their respective tasks. These large-scale datasets are typically scraped from the web, and thus algorithms trained on this data could be expected to perform well on other web-scraped datasets [3]. However, the images available on the web are still significantly different from the images captured by cameras deployed in real-world contexts. Consequently, benchmarks on large-scale datasets cannot be expected to translate to real-world data [3, 24]. That is not to say that performance on benchmark datasets can be disregarded, they still provide insights into comparative performance between algorithms. Furthermore, algorithms trained on large-scale datasets have been shown to improve performance when transferring to target domains with limited available training data, compared to models trained solely on data from the target domain. However, this benefit has shown to decrease as the number of examples in the target domain increases [11].

Deploying a computer vision algorithm in a real-world environment presents several challenges that can impact the algorithm's robustness and accuracy. Some of these challenges are a direct result of a dynamic environment that varies widely in lighting conditions, weather conditions, and viewing angles. The quality of the images also varies greatly depending on the camera used to capture them. Even though two cameras can produce images that have the same resolution, bit-depth, or file size, the quality of the resulting image can vary dramatically due to sensor-induced noise and artifacts. Furthermore, real-world scenarios can also introduce heavy occlusions and unexpected object configurations, making tasks such as object detection and segmentation increasingly challenging.

2 Leveraging Continuous Data-streams

Traditional methods in computer vision often involve breaking down the continuous flow of data captured by visual sensors into individual frames, subsequently processing these frames in a sequential manner [28, 29]. However, the use of temporal cues from sequences of frames constitutes a transformative approach that can significantly improve the capabilities of computer vision systems. The integration of motion cues, contextual cues, and long-term relationships between objects within a scene introduces a new avenue for understanding and interpretation [9, 22]. Motion cues inherently capture the dynamic nature of real-world scenarios, enabling algorithms to discern movement patterns, trajectories, and interactions. Whereas, contextual cues offer a broader understanding of the context, enriching the interpretation of object behaviors within the scene. Long-term relationships established across the temporal dimension allow for the modeling of intricate dependencies, unveiling complex understanding that cannot be represented by a single image. The introduction of fully attentional models, in particular transformers, has expanded the capabilities of vision systems to effectively model long-range relationships across any available dimension. Since their introduction to the visual domain in 2020 [1, 2], they have shown impressive capabilities for most vision tasks [7, 10, 15, 20, 23, 26].

3 Concept Drift and Contextual Awareness

Computer vision systems are deployed over longer periods in real-world environments and are exposed to gradual and sudden changes in visual appearance as a result of environmental conditions. As the appearance of objects changes so does the visual concept that describes them, this is known as concept drift. Concept drift poses a significant challenge as it can result in unpredictable behavior of the vision system, which in turn would make it unreliable, and would require ongoing quality assurance which can be costly [8, 16, 30]. In tasks such as autonomous driving, medical image analysis, security, and food inspection, vision systems need to be reliable as a mistake could result in serious harm. Concept drift induces a change in the underlying data distribution which changes the statistical properties of the data distribution. This change may lead to the algorithm's learned patterns and assumptions becoming outdated and less effective [17, 25, 27]. Addressing concept drift is a challenging problem as it is difficult to detect and estimate the magnitude of the drift, to adapt to the changes in an informed way. Furthermore, to evaluate the efficacy of a given adaptation method, labeled data is required to determine the performance impact of the observed concept drift [17, 27, 30]. The impact and type of concept drift observed are dependent on the context and sensor

used to capture the data, making it difficult to directly compare methods. Moreover, existing datasets also lack rich meta-data that could help identify parameters that induce drift. Although it would be very difficult to identify every drift-related parameter, having a contextual understanding might provide clues that can guide intelligent adaptation. Weather-related parameters could be pulled from meteorological sensors and used in conjunction with a vision system [6]. Such a weather-aware system could then learn to relate the drift in visual appearance to specific weather conditions, effectively allowing the system to anticipate drift and learn how to address it.

In summary, the synthesis of ANNs and camera technologies have provided us with artificial eyes and brains that can mimic human perception to a remarkable extent. These advancements have led to notable achievements in various vision tasks. Ranging from understanding dynamic motion patterns to capturing intricate spatial relationships, and detailed scene understanding. However, the transition from controlled research settings to a dynamic and unpredictable real-world setting introduces a wide array of unique challenges. These challenges include the inherent limitations of available labeled data, handling contextual variations, addressing concept drift, and ensuring robustness in the face of occlusions, and diverse lighting conditions. Raising a crucial question: *how we can effectively train and adapt these systems to perform consistently and reliably in the intricate and often unpredictable landscapes of the real world?*

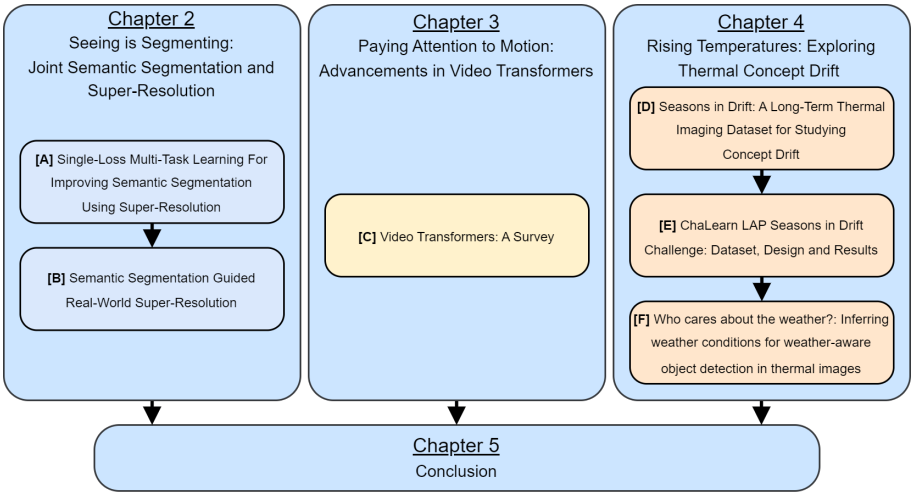


Fig. 1.2: In this figure an overview of the four main chapters of the thesis can be seen. Each chapter and sub-topic is visualized as its own box. Arrows between boxes denote an extension of the prior topic.

4 Thesis Structure

This thesis is divided into four main chapters, followed by a collection of papers. In the following chapters, we will discuss methods, datasets, and evaluation of real-world applications of object detection and segmentation algorithms.

Chapter 2 **Seeing is Segmenting:**
Joint Semantic Segmentation and Super-Resolution: Aims to investigate the relationship between input resolution and performance of semantic segmentation. Specifically, focusing on employing super-resolution as a way to recover lost information from low-resolution images, investigating the symbiotic relationship between super-resolution and semantic segmentation. In this chapter we will discuss and detail how jointly learning semantic segmentation and super-resolution can improve the performance of either task and enable semantic segmentation systems to be used on low-resolution camera feeds.

Chapter 3 **Paying Attention to Motion:**
Advancements in Video Transformers: Aims to investigate the advent of transformers for modeling video. Specifically, focusing on the challenges posed when processing high-dimensional visual data, such as videos, and the trends and techniques employed to reduce the computational burden and handle the vast information redundancy introduced with video data. We will provide an overview of architectural changes and methods observed in the literature, and summarize key insights relating to video transformers as a whole as well as insights directly related to object-centric tasks.

Chapter 4 **Rising Temperatures:**
Exploring Thermal Concept Drift:
Aims to investigate the impact of thermal concept drift in long-term deployments of computer vision systems. Specifically, analyzing the correlation between performance and drift-inducing factors, evaluating the performance of thermal object detection under concept drift, and weather-aware conditioning through a fine-grained auxiliary optimization task impacts the performance of thermal object-detection algorithms. This chapter will detail and discuss the impact of thermal concept drift on vision tasks (namely, object detection and anomaly detection), introduce a novel dataset created to facilitate further research into concept drift in the thermal domain, and discuss methods for conditioning object-detection algorithms to learn weather-aware representations.

Chapter 5 **Conclusion**: Summarizes and concludes the key findings of this PhD thesis.

References

[1] 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: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] A. Fang, S. Kornblith, and L. Schmidt, "Does progress on imagenet transfer to real-world datasets?" *arXiv preprint arXiv:2301.04644*, 2023.

[4] H. Gajjar, S. Sanyal, and M. Shah, "A comprehensive study on lane detecting autonomous car using computer vision," *Expert Systems with Applications*, p. 120929, 2023.

[5] Y. Gao, M. Zhou, and D. N. Metaxas, "Utnet: a hybrid transformer architecture for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*. Springer, 2021, pp. 61–71.

[6] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 305–310.

[7] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," in *IEEE TPAMI*, 2022.

[8] A. S. Iwashita and J. P. Papa, "An overview on concept drift learning," *IEEE access*, vol. 7, pp. 1532–1547, 2018.

[9] S. Jenni, G. Meishvili, and P. Favaro, "Video representation learning by recognizing temporal transformations," in *ECCV*, 2020.

[10] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM CSUR*, 2022.

[11] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?" in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2661–2671.

[12] 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.

[13] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov *et al.*, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, 2020.

[14] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[15] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, "A Survey of Visual Transformers," *arXiv*, 2021.

[16] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[17] C. Mera, M. Orozco-Alzate, and J. Branch, "Incremental learning of concept drift in multiple instance learning for industrial visual inspection," *Computers in Industry*, vol. 109, pp. 153–164, 2019.

[18] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges," in *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 2018, pp. 35–38.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICAI*, 2015, pp. 234–241.

[20] L. Ruan and Q. Jin, "Survey: Transformer based video-language pre-training," *AI Open*, 2022.

[21] 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.

[22] F. Sener, D. Singhania, and A. Yao, "Temporal aggregate representations for long-range video understanding," in *ECCV*, 2020.

[23] A. Shin, M. Ishii, and T. Narihira, "Perspectives and prospects on transformer architecture for cross-modal tasks with language and vision," *IJCV*, 2022.

[24] D. Tsipras, S. Santurkar, L. Engstrom, A. Ilyas, and A. Madry, "From imagenet to image classification: Contextualizing progress on benchmarks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9625–9635.

[25] Q. Xiang, L. Zi, X. Cong, and Y. Wang, "Concept drift adaptation methods under the deep learning framework: A literature review," *Applied Sciences*, vol. 13, no. 11, p. 6515, 2023.

[26] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, "Transformers in computational visual media: A survey," *Computational Visual Media*, 2022.

[27] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "{CADE}: Detecting and explaining concept drift samples for security applications," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[28] L. Zhou, C. Xu, and J. Corso, "Towards automatic learning of procedures from web instructional videos," in *AAAI*, 2018.

[29] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8856–8865.

[30] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," *Big data analysis: new algorithms for a new society*, pp. 91–114, 2016.

Chapter 2

Seeing is Segmenting: Joint Semantic Segmentation and Super-Resolution

1 Introduction

In the field of computer vision, the acquisition and analysis of visual data form the foundation for several critical applications, spanning from medical image analysis [20, 36, 64] and satellite imagery [34, 37, 66] to autonomous vehicles [9, 19, 50, 54] and surveillance systems [4, 22, 39, 71, 83]. A key component of these tasks is the ability to resolve intricate details from the captured images which requires a certain level of image fidelity. Thus the resolution of a given image plays a vital role in computer vision systems. Higher image resolution translates to enhanced accuracy, finer granularity, and improved perceptual quality of the visual content. However, this ideal scenario is often hindered by practical limitations, such as hardware constraints. The size of the camera sensor, communication bandwidth, compression, etc. all impact the resolution of data that can be transmitted and/or stored. Low-Resolution (LR) and compression artifacts tend to negatively impact the performance of various vision tasks [23, 66], and thus become a restrictive factor deployment of automated vision systems in real-world applications.

In the field of Super-Resolution (SR) the aim is to intelligently restore the finer details of LR, thus obtaining a more detailed High-Resolution (HR) version of the original image [5, 40, 42, 91]. The rationale underlying SR is grounded in the belief that augmenting the fidelity of imagery can positively benefit downstream vision tasks by recovering finer details, enhancing object boundaries, and facilitating more accurate feature extraction. In simple terms,

SR can be seen as a reconstruction task where resolution is restored, and neighboring information is employed to infer missing details. Prior research in SR has shown that it greatly improves other vision tasks when used as a preprocessing step [14]. However, these advantages are reliant on the assumption that the HR representation preserves relevant contextual cues for scene interpretation. This prompts an interesting question: can SR be leveraged in a symbiotic partnership with other vision tasks, surpassing the individual task's limitations and amplifying the strengths of both?

To accurately reconstruct high-frequency details, a robust understanding of the scene's semantics is crucial to understand underlying patterns for a given region. Granular scene understanding also plays a significant role in other vision tasks, such as Semantic Segmentation (SS) [60]. At its core, SR aims to recover the latent details hidden within low-resolution images, whereas SS seeks to partition the image into coherent regions, associating each pixel with a specific semantic label that reflects the underlying objects and structures. Therefore, the joint learning of SS and SR presents a captivating area of research, where the fusion of spatial precision and contextual semantics could prove beneficial and present novel insights in the visual domain.

This chapter serves to provide insight into multi-task learning of Real-World Super-Resolution (RWSR) and Semantic Segmentation (SS) by addressing approaches, namely:

Paper A **Multi-task Learning via Single-task Optimization**: Jointly-learning SR directly from a SS training loop, without additional SR based optimization. Allowing SR algorithms to train domain-specific knowledge following a Real-World Super-Resolution (RWSR) context.

Paper B **Semantic Guidance of Super-Resolution for Real-World Applications**: Guiding SR to obtain semantically rich representations, through guided learning of a parallel SS branch. Improving the perceptual quality of super-resolved images without any additional computational cost at test-time.

2 Background

Generally in computer vision, the ability to extract meaningful information from images and video is a crucial step to obtain robust and accurate systems [7, 78]. To accurately extract fine-grained information resolution is often a key factor [66, 71, 83]. This is particularly true for low-level tasks that make fine-grained predictions [13, 14].

2.1 Single-Image Super-Resolution

SR can roughly be divided by approaches that use multiple images (i.e., multi-view or temporally aligned) and single image SR. Both are viable solutions for recovering detail in images. However, the work presented in Paper A and Paper B focuses on joint learning of SR and SS. As such the scope is limited to learning-based single-image SR. As such recovering those details can be crucial for real-world applications with LR data streams.

SR is a well-researched topic in computer vision, and have been extensively studied since 1974 [21]. Much like many other computer vision tasks, the introduction of Deep Learning (DL) revolutionized the field [16]. In its simplest form, single-image SR is a mapping between a LR domain, and a HR domain. Following this intuition, traditional learning strategies involve synthetically degrading and down-sampling HR images to create HR-LR pairs [16, 42, 48, 75, 91]. The models are then trained to accurately reconstruct the original HR image, typically by minimizing Mean Squared Error (MSE). Similarly, the evaluation typically focuses on optimizing Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity index (SSIM), despite recent research having shown that these metrics fundamentally disagree with human observers [1, 40, 81]. The introduction of Generative Adversarial Networks (GANs) in SR presented a perceptual component to SR, where a competing discriminator network had to be unable to distinguish between the generated HR and a real HR image [40]. This forces the network to reconstruct finer details that mimicked real images, resulting in images of higher perceptual quality [40, 81]. While this approach resulted in images that were more realistic when inspected by humans, the reconstructed details weren't accurate, and often resulted in lower PSNR. To address this learned metric were proposed to act as a proxy for human perceived perceptual quality [15, 17, 41, 90, 95].

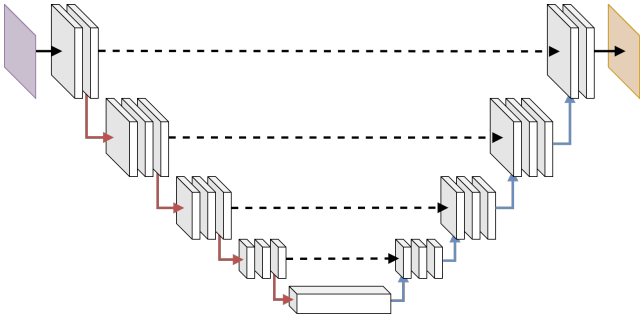
Recent work has highlighted that traditional approaches essentially attempt to create an inverse mapping of the degradation function rather than the noise that would be observed in a real-world context [18, 33]. In a real-world context, the degradation consists of an unknown set of degradations, which are poorly represented by traditional synthetic pairs [5, 33]. The field of Real-World Super-Resolution (RWSR), aims to reconstruct images without a known HR ground-truth [33, 46, 48, 53, 67, 67, 70, 70]. RWSR is inherently an ill-posed problem as there is no way to accurately assess how faithfully the image was reconstructed. Thus the performance of these metrics is commonly evaluated with qualitative comparison [33, 53, 79, 81], and in some cases report traditional metrics (i.e., PSNR, SSIM and Learned Perceptual Image Patch Similarity (LPIPS)) for prosperity. Initial methods of RWSR aimed at addressing this problem with a zero-shot approach, where degradation kernels are estimated at test-time and then applied to reconstruct the high-resolution image [5, 48, 67, 70]. Alternately, RealSR [33] proposed extracting blur kernels

and noise patches from real images to synthetically create 'realistic' LR images, then used a combination of adversarial-, distance-, and perceptual-losses to approximate quality during training. RWSR models' ability to reconstruct details in real-world images presented an exciting avenue for supporting other vision tasks [13, 14]. Notably SR models helped improve the performance of optical character recognition by up to 15% [62] and object detection in satellite images by up to 30% [66].

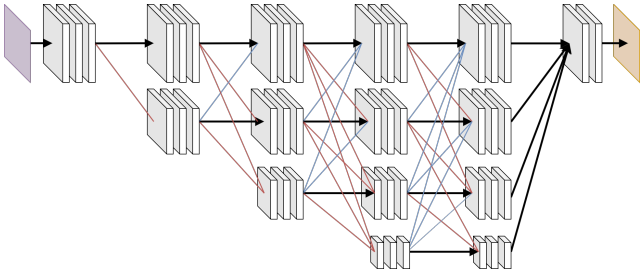
2.2 Semantic Segmentation

SS is a widely studied computer vision problem focused on dense scene understanding by assigning per-pixel labels. SS is commonly used in autonomous driving [9, 19, 50, 54], medical image analysis [20, 36, 64], and robotic sensing [6, 51, 68] where fine-grained scene-understanding is a crucial component of the vision system. Early segmentation systems used contours, edges, and hand-crafted texture descriptors to segment images [30, 82], and then clustering to assign a class label. Similarly to SR and other vision tasks, the introduction of DL in computer vision fundamentally changed SS. Convolutional Neural Networks (CNNs) ability to extract semantic understanding for image-classification [25, 38, 69, 72] showed the great potential of CNNs for vision tasks. However, CNNs tends to progressively increase the receptive field to obtain semantic understanding from the entire image at the cost of spatial resolution. To perform pixel-wise classification, recovering spatial resolution is a crucial component.

U-Net [64] (visualized in Figure 2.1a), proposed an architectural scheme that would progressively recover the spatial resolution from the encoding CNN, while also leveraging skip-connections to propagate early representations retaining high spatial detail, which subsequently was combined with the up-sampled representation through convolutional layers [64]. The U-Net style of architecture has become the standard for many SS approahces [3, 20, 32, 43, 64, 87]. Regrettably, these architectures fail to capture semantically meaningful representations at earlier layers [57, 77], thus relying on the up-sampling step to learn a spatial re-mapping. By continuously fusing information from earlier layers to recover spatial detail, U-net obtained an increase in performance and notably became significantly more accurate at pixels bordering two classes [57, 77]. HRNet (visualized in Figure 2.1b), further refined this by leveraging multiple parallel branches which continuously share information by progressively infusing higher branches with semantic information, while infusing lower branches with spatial detail [77]. Variants of HRNet also showed great performance in Pose-Estimation [11, 77], Optical Character Recognition [73, 85] and Object-Detection [58].



(a) UNet [64]



(b) HRNet [77]

Fig. 2.1: Schematic overview of the UNet [64] and HRNet [77] architectures. Purple represents the input tensor, orange represents the semantic predictions output of the system, and black lines represent the general flow of data. Red lines represent a loss in spatial resolution, whereas blue lines represent an increase in spatial resolution. the dotted lines represent residual connections. For readability fusion connections spanning different branches in Figure 2.1b have had their arrows removed, but follow the general dataflow, i.e., left → right

2.3 Multi-Task Learning

Many computer vision tasks have shown to benefit initializing from models pre-trained for other tasks, typically image classification [24, 28], implying the existence of patterns and semantic understanding which are transversal across tasks. Multi-task learning further extends on this by leveraging parts of a network to perform multiple tasks simultaniously [32, 56, 65, 92, 93]. Not only is it computationally efficient, it has also shown to improve performance of the individual tasks [49, 65, 84]. Multi-task learning approaches generally tend to employ a shared feature-extraction backbone or encoder network, which then feeds task-specific branches [26, 56, 60, 74]. The intuition is that even when optimized separately with task-specific losses, the combination of losses will guide the network to reach a local minimum which is otherwise difficult to reach when optimizing an individual task [60, 74]. As many tasks differ in training data, i.e object-detection requiring instance annotation, while tracking additionally requires temporal correlation, a combination of tasks that can leverage similar training data is often employed [26, 27, 32, 60]. While simultaneous optimization of these tasks is by far the most common, recent works have also employed approaches that switch between data and task optimization in a cyclical manner [29, 49, 74]. Typically the performance of each task is evaluated on datasets that are common for their respective tasks providing a general indication that multi-task learning is beneficial for many vision tasks.

3 Multi-task Learning via Single-task Optimization

Generally in SS, the performance of the system is highly correlated with the input resolution., particularly when segmenting small objects [79]. This is particularly troublesome in contexts where the vision system needs to accurately perceive distant objects, as both camera quality and the size of objects degrade the performance of the given system. This presents a restrictive problem where information is inherently sparse, and lacks detailed information to accurately obtain a semantic understanding of the scene. Thus, employing SR networks to increase the resolution of the input and recover fine details poses an interesting avenue for study. Furthermore, incorporating the SR network directly into the pipeline could allow for context-specific semantic understanding as well as refining underlying patterns that benefit the primary task.

3.1 Related Works

Previous work in joint learning of SR and SS tend to leverage either task as an auxiliary optimization goal which is discarded at test time [60, 79]. Namely, [60]

proposes leveraging an auxiliary prediction head to perform semantic segmentation, thus the majority of the network is shared between the SR and SS heads. Intuitively this approach enforces the network to include some level of semantic understanding in intermediate representations [60]. Notably, segmentation at the border of classes remains a challenge for the proposed method, leaving them to employ a novel boundary mask to avoid forcing the SR head to focus on class boundaries. Inversely, DSRL [79] proposes using SR as an auxiliary task to improve SS. Namely, they propose to guide intermediate representations of the SS module to generate high-resolution representations by minimizing the SSIM between intermediate features of the SR module and SS module respectively [79]. Notably, DSRL [79] saw improvement in semantic segmentation across multiple architectures and particularly exhibited increased accuracy of boundary pixels. Recent work performing joint SS and SR still rely on LR-HR pairs for optimization of the SR module, which is ill-suited for contexts where HR is unavailable.

3.2 Improving Semantic Segmentation using Super-Resolution

SS of LR input remains a challenge that could greatly benefit from joint multi-task learning of SR and SS. This is further underlined by previous work that shows SR- and SS-networks can effectively share intermediate representations and excel in their respective tasks [26, 60]. In addition to the trend of SR guided SS [79], the SS task could potentially be used as a viable proxy for SR. Thus, jointly learning both tasks through the optimization of a single task should be possible.

Multi-Task Semantic Segmentation and Super-Resolution

Following this intuition Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR) was proposed as a way to jointly learn SR and SS in an end-to-end manner. While existing methods focus on optimizing the respective tasks separately and guiding each prediction head towards its own global optimum, the purpose of MT-SSSR is solely to improve the performance of the downstream task (i.e., segmentation). Thus allowing the SR module to learn without any LR-HR pairs, effectively adapting it to a RWSR approach. As shown in Figure 2.2, the construction of MT-SSSR is rather simple, and aimed to keep the SR and SS modules interchangeable with other architectures from their respective tasks. Architecturally the SR module is completely separated from the SS network, thus as long as the output is a tensor of corresponding spatial resolution to the input, the choice of SR module should be arbitrary. The generator module of ESRGAN [81] was chosen for its ability to reconstruct high-frequency details, and its impressive performance on various datasets [55, 61, 81]. Furthermore, it is crucial to retain high-resolution representations throughout the

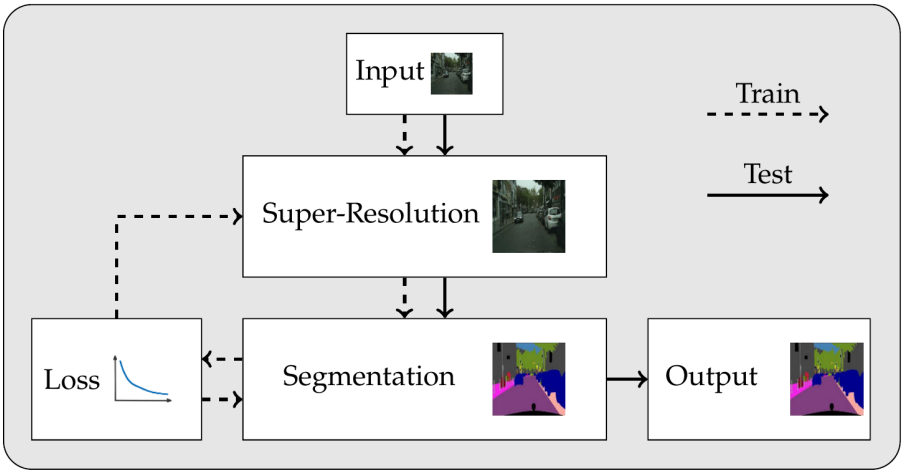


Fig. 2.2: *"Our proposed framework, Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR). Dashed and full lines represent training and testing phases, respectively. The SR model learns to upsample and enhance the input image based on the segmentation task loss. The segmentation model uses the same loss to improve the accuracy of its prediction."* [1]. This figure was adapted from [1], (Paper A)

network, especially for LR inputs [57, 77], thus HRNet [77] was selected as the SS module, due to its ability to retain HR representations throughout the network [73, 77].

Naturally one of the drawbacks of this style of multi-task learning is the increase in computational cost, as a result of performing inference of two DL. The resulting complexity could be computationally restrictive in certain contexts.

Experiments

Training and evaluation of the proposed framework were conducted on Cityscapes [12] as well as IDD20k-Lite [52] (a LR version of [76]). Cityscapes consists of driving scenes in urban environments recorded in 50 different cities, whereas IDD20k-lite consists of data ranging from urban areas to rural suburban areas of cities in India. These datasets are both diverse in terms of classes, but also significantly different in terms of textures. This configuration made it ideal for inspecting if the framework would generalize to distinct types of data.

As leveraging pre-trained models trained on large-scale datasets has shown consistently beneficial, even for training in distinctly different domains [24, 28], both the SR and SS modules were initialized from pre-trained models. The SS module was initialized from a model pre-trained on COCO [44], whereas the SR module was initialized from a model pre-trained on DF2K [2, 81] (a combination of DIV2k and Flickr2K).

Furthermore, an ablation study was conducted to evaluate the performance of MT-SSSR in the traditional synthetic LR-HR setting. Additionally, comparisons with bicubic interpolation and single-task optimization of the SR module were also included in the ablation study. Note that single-task refers to the SR module being trained in a traditional fashion described in the original paper [81], as opposed to the proposed MT-SSSR framework. Additionally, the baseline HRNet was extended, by leveraging a State-of-the-Art (SotA) alternative to cross-entropy loss, namely Region Mutual Information (RMI) loss. RMI loss aims to use the statistical properties of the nearby region to dynamically adjust the loss based on how difficult the given pixel is to predict, given its neighbors.

Due to memory constraints imposed by such a large model, training adopted a cyclical approach where batches alternated between cropped patches of 128×128 patches and full images. During the patch phase, both SR and SS module was updated, however, during full-image phases only the SS module was updated.

3.3 Results and Insights

To evaluate the performance of the system, the SS network is evaluated using the common mean Average Precission (mAP) metric. mAP in semantic segmentation measures the mean of the Average Precission (AP) of each class. As shown in Table 2.1, the performance of the proposed method beats SotA by a significant margin on both Cityscapes (i.e., 3.6%) and IDD-Lite (i.e., 2.5%). Additionally, a significant improvement over the stand-alone HRNet is also observed across both datasets.

Cityscapes (Real LR images)

Method	Size	Valid. mAP	Test mAP
DeepLabV3+ [8]	×1	0.700	0.671
PSPNet [86]	×1	0.715	0.691
HRNet [77]	×1	0.773	0.754
DSRL [79]	×2	0.757	0.748
MT-SSSR (ours)	×2	**0.803**	**0.790**

(a)

IDD (Real LR images)

Method	Size	Valid. mAP
DeepLabV3+ [8]	×1	0.643
ERFNet [63]	×1	0.661
HRNet [77]	×1	0.694
Eff-UNet [3]	×1	0.738
MT-SSSR (ours)	×2	0.741
MT-SSSR (ours)	×4	**0.763**

(b)

Table 2.1: Performance of the SS module on the Cityscapes validation and test sets (table 2.1a) as well as the IDD validation set (Table 2.1b). Best performing models are highlighted in **bold**. This table is adapted from [1], (Paper A).

Cityscapes (Real LR images))

Method	Size	mAP
HRNet [77]	×1	0.773
HRNet + RMI	×1	0.774
HRNet + RMI	×2 Bicubic	0.780
HRNet + RMI	×2 SR_{ST}	0.781
MT-SSSR (ours)	×2 **SR_{MT}**	**0.803**

(a)

IDD (Real LR images)

Method	Size	mAP
HRNet [77]	Native	0.694
HRNet + RMI	Native	69.9
HRNet + RMI	×2 Bicubic	0.709
HRNet + RMI	×4 Bicubic	0.671
HRNet + RMI	×2 SR_{ST}	0.712
MT-SSSR (ours)	×2 SR_{MT}	0.741
MT-SSSR (ours)	×4 **SR_{MT}**	**76.3**

(b)

Table 2.2: This table shows the ablation study of each module included in the MT-SSSR framework on the Cityscapes (Table 2.2a) and IDD (Table 2.2b) datasets. This figure compares the impact of bicubic interpolation, traditional SR (i.e., SR_{ST}) and the complete MT-SSSR framework (i.e SR_{MT}).Best performing models are highlighted in **bold**. This table is adapted from [1], (Paper A)

Furthermore, as shown in Table 2.2, simply interpolating the input provides an increase in segmentation accuracy, leveraging a SR module will further improve the performance. Additionally leveraging a multi-task scheme like MT-SSSR can further increase the performance of the SS module. In Table 2.2 it is further outlined that this trend persists for up-sampling by a factor of two and four.

Fig. 2.3: Examples of LR input images (top row) and the corresponding super-resolved HR counterparts (bottom row), produced by MT-SSSR. This figure is adapted from [1], (Paper A)

A concern with the MT-SSSR framework, is that solely leveraging the SS task for optimization could force the SR module to act as a traditional encoder network, which provides semantically meaningful representations, without guaranteeing the 'super-resolved' image retaining the visual information of the input. In Figure 2.3 it is shown that the super-resolved solved image

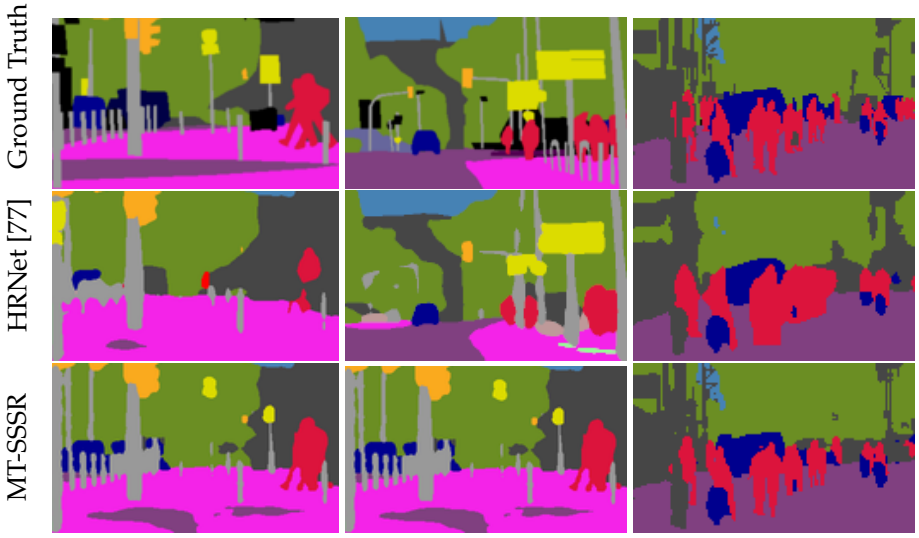


Fig. 2.4: Examples of ground-truth segmentation maps (top row), baseline HRNet semantic predictions (middle row), and MT-SSSR semantic predictions (bottom row). This figure is adapted from [1], (Paper A)

still retains the visual appearance of the input, recovers fine-grained details and increases the subjective perceptual quality of the image. Interestingly all images, images produced by this MT-SSSR framework tend to have a slight blue hue and sharper borders at objects. The borders can be explained by the nature of SS benefiting from clearer delineations between objects, while the shift in hue is an unexpected change. Intuitively this can be assumed to be the SR network trying to shift all images toward a distinct color palette, simplifying the work for the SS module

When inspecting the segmentation masks (as shown in Figure 2.4), it can further be observed that MT-SSSR more faithfully reconstructs the fine detail of small and thin objects, such as legs, traffic lights, and in some cases entire people. The increased performance and accuracy can thus be observed to recover fine-grained information that could be vital for interacting with the real world. In some cases, the resulting increase in computational complexity could be restrictive, which potentially could be addressed by leveraging lightweight SR and SS modules.

3.4 Summary and Contributions

Input resolution remains a vital component for the segmentation task, and while increasing resolution via interpolation is present as an easy improvement, the improvement is insignificant. In this section, a novel multitask

framework was presented which addresses join learning of SS and SR. The proposed method jointly optimizes both tasks solely from the SS loss and surpasses separately training a SR module in terms of mAP on both Cityscapes and IDD-Lite. Furthermore, the proposed MT-SSSR framework allows the SR module to be trained on real-world LR data.

In this section, the work conducted in Paper A was discussed. In large parts, the work focused on joint-multitask learning of semantic segmentation and super-resolution, for the purpose of increased performance of semantic segmentation models. As well as the impact on fine-grained segmentation masks as well as the changes observed within the SR network. The contribution of the work described herein can be summarized as follows:

- The novel Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR) framework was proposed, which significantly improves the performance of existing segmentation models, by jointly learning super-resolution and semantic segmentation with interchangeable super-resolution and segmentation networks.
- The proposed MT-SSSR network only requires segmentation labels, thus making it applicable in RWSR contexts where LR-HR pairs aren't available.
- The proposed method achieved SotA semantic segmentation performance on challenging LR variants of Cityscapes and IDD20K datasets.

4 Semantic Guidance of Super-Resolution for Real-World Applications

Real-world applications of super-resolution rely on the reconstruction of lost detail, which is an inherently ill-posed problem. Particularly as the LR input could be generated from any number of HR ground truths due to the highly complex degradation undergone during capture. Recent work attempts to address this through domain adaptation of pre-trained models [33, 46, 94] or zero-shot models that approximated degradations at test-time [5, 70]. Which relies on accurately estimating degradations from the target domain, which is challenging to do in a generalizable manner, and cumbersome if done at test time.

Inspired by the observations in Paper A [1] and other works [60, 79], indicating that guidance from a SS network can be beneficial for obtaining sharper object boundaries and reduced noise. Thus the task of SS could be used as an optimization proxy that does not require known HR ground truths. However, the resulting images shown in Figure 2.3 and described in Paper A contain

undesirable color changes, which potentially also affect learned textures [29]. Thus additional methods must be employed to color-correct reconstruction.

4.1 Related Work

RWSR approaches assume LR image is a product of blur kernel k, scaling - factor s with an added noise pattern applied to its corresponding HR counterpart, i.e $I_{LR} = (I_{HR} * k) \downarrow_s + n$. Thus to effectively reverse the degradation, blur-kernel, and noise needs to be estimated [5, 67, 70]. However, despite showing great performance on RWSR benchmarks, this simplifies the problem to an extent that fails to approximate the complex degradation process [5]. To combat this KernelGAN [5] proposed additionally incorporating a perceptual loss to guide the network toward generating more realistic-looking images, underlining that using auxiliary losses can help guide RWSR models.

This extends to previous work on the guidance of image-generation tasks. Where it has been shown that including semantic understanding can greatly impact the type of textures generated [10, 31, 97]. A trend which translates to the image-reconstruction domain [45, 59, 60, 80, 89]. Particularly semantic conditioning at the feature level has been shown to help with generating more realistic and semantically appropriate textures [59, 80] for SR tasks. However, generating semantic labels to condition the network at test time can be very computationally prohibitive, to address this existing work adapt guidance approaches that can be entirely discarded at test-time [60, 79, 89]. Typically these approaches share the encoder part of the network so that intermediate representations can be guided by the auxiliary task. Notably [89] proposes enforcing structural similarity at the last decoder level of the different tasks, which helps sharpen the borders and recover small objects, and the nature of the separated decoders enable interchangeable SotA CNN models for either task. Furthermore, the model shares a decoder network but employs task-specific decoders for the shared representation. Consequently, the guidance of the SR module to learn semantically meaningful representations is not ensured. While these approaches allow each task to extract task-specific features from a shared backbone network, the performance of the system is very sensitive to the weighting of the auxiliary heads [89]. Furthermore, recent works still leverage synthetic LR-HR pairs for training the super-resolution model, which fails to fully capture the degradation observed in RWSR.

4.2 Semantic Segmentation Guided Real-World Super-Resolution

To address the shortcomings of recent work and enable learning semantically meaningful representations without sacrificing the accurate reconstruction of the input image the SSG-RWSR framework was proposed. SSG-RWSR aims to enable the training of RWSR models through semantic guidance and

domain adaptation to produce more accurate and noise-free HR images. SSG-

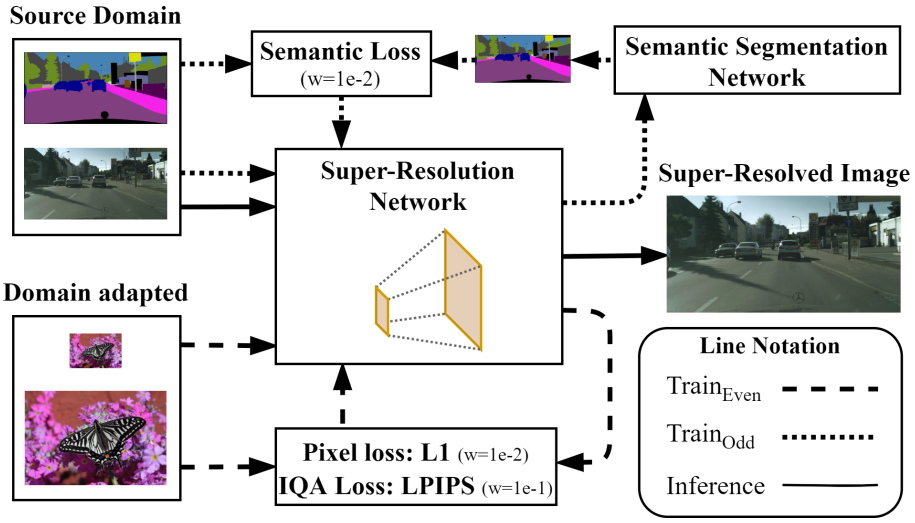


Fig. 2.5: *"Schematic overview of our proposed SSG-RWSR. To learn to perform RWSR we leverage both guiding from an auxiliary semantic segmentation task and domain adaptation. At test time, the semantic segmentation network is de-coupled, and as such no semantic labels are required to super-resolve the LR test images."* [29]. This figure is adapted from [29], (Paper B).

RWSR achieves this through a cyclical training loop (visualized in Figure 2.5), extending on the work detailed in Section 3 with a domain-adaptation branch. As outlined in Section 3, SS models benefit from inputs with low levels of noise and fine details, and thus can serve as a proxy for optimizing image quality. Contrary to related work [60, 79, 89], the work conducted in Paper A indicated that directly linking the two networks is more beneficial than treating them as parallel tasks To ensure LR-HR image consistency and high-frequency detail, a domain-adaption method [33] was included to learn real-world degradation on large-scale HR data. The SR model is thus cyclically trained, by alternating between semantic guidance and domain-adaption.

Semantic Guidance

The semantic guidance originally proposed in Paper A, is directly applied in the semantic guidance cycle. During this phase the SR network and SS network sequentially process the input LR image. The segmentation loss is then back-propagated to update both SR and SS modules. If the SS module remained frozen, the resulting semantic guidance would likely reinforce patterns and biases, from its original training data. Thus the discovery of new patterns as a result of improved SR would be suppressed. During semantic guidance, the training data consists of data from the desired source domain.

Domain Adaptation

As observed in Paper A, SR networks guided purely by SS result in changes to the image which are undesirable for image-reconstruction tasks. Thus to prevent such influence traditional synthetic LR-HR training pairs are employed. However, blindly using these pairs for training would result in the model trying to model two domains at once (i.e., source $domain \rightleftharpoons texture$ domain) potentially preventing accurate generalization in either domain [96]. To address this, blur kernels and noise patches are extracted from the source domain, creating a pool of realistic blur kernels and noise patches. During training a random blur-kernel and noise patch is selected to degrade a given HR sample, thus creating synthetic LR-HR pairs that exhibit similar degradation to that of the source domain, thus reducing the $texture \rightarrow source$ domain gap.

The combination of semantic guidance and domain adaptation facilitates training of RWSR in a target source domain that lacks HR ground truths. Furthermore, at test time the auxiliary tasks can be entirely discarded, meaning the proposed SSG-RWSR framework has no impact on inference at test time.

Experiments

The experiments conducted to evaluate the performance of the proposed framework were twofold: Firstly, focusing on the evaluation of 'real' images, i.e., images where the HR ground-truth is unknown, and secondly, focusing on the reconstruction of synthetically degraded images following traditional SR evaluation protocol. Evaluation of real images was done on Cityscapes [12] and IDD20K [76], as both contained semantic labels, allowing for semantic guidance. As there exist no HR ground-truths for these datasets, it is impossible to employ traditional metrics such as PSNR, SSIM. To get a notion of performance two perceptually focused metrics which show a good correlation with human judgment. Namely, Neural Image Assessment (NIMA) [17] and Meta Image-Quality-Assesment (Meta-IQA) [95]. Additionally Mean Opinion Rank (MOR) is measured as a direct measure of human perceived perceptual quality [47].To make the framework comparable with traditional SR metrics, additional LR-HR pairs are generated from the cityscapes dataset. Thus enabling direct SR quality metrics (i.e., PSNR and SSIM). Furthermore, the performance is also measured using MOR and perceptually focused metrics LPIPS [90], and Deep Image Structure and Texture Similarity (DISTS) [15] Meta-IQA [95], and NIMA [17]

The proposed method is compared to four SotA methods designed for SR of real-world images. Namely, MZSR [70], DPSR [88], RealSR [33], and DAN [48]. As most of competing models require configuration towards the target domain, adjustments were made for each model following the protocol outlined in their respective papers [5, 48, 70, 88].

4.3 Results and Insights

Bicubic RealSR [33] DPSR [88] DAN [48] Ours [29]

Fig. 2.6: *"Comparison with SotA methods for ×4 of **synthetically** degraded images from the Cityscapes dataset. As visible, our method reconstructs sharp images with low noise compared to the existing methods."* [29], This figure is adapted from [29], (Paper B).

Bicubic RealSR [33] DPSR [88] DAN [48] Ours [29]

Fig. 2.7: *"Comparison with SotA methods for ×4 SR of **real** images from the Cityscapes dataset. As visible, our method reconstructs sharper and more visually appealing results compared to the existing methods."* [29]. This figure is adapted from [29], (Paper B).

As can be observed in Tables 2.3 and 2.4 the proposed framework (i.e., SSG-RWSR) produces images that surpass the perceptual quality of SotA models according to MOR, NIMA and Meta-IQA. This is further underlined by the NIMA and Meta-IQA scores, where the proposed method is only slightly surpassed by DAN on the IDD Dataset. Evaluating perceptual quality, particularly one correlated with human quality assessment, is difficult to relate to from num-

Cityscapes (Real LR images)			
Method	NIMA↑	Meta-IQA↑	MOR↓
Bicubic [35]	4.62	0.245	-
ESRGAN [81]	4.95	0.247	-
MZSR [70]	4.88	0.231	3.33
DPSR [88]	4.83	0.240	4.41
RealSR [33]	4.87	0.236	2.75
DAN [48]	4.65	0.246	3.47
Ours	**5.04**	**0.254**	**1.21**

Table 2.3: *"Quantitative results on the Cityscapes validation set. ↑ and ↓ indicate whether higher or lower values are desired, respectively. As seen, our method obtains both the best MOR and NIMA and Meta-IQA results."* [29]. This table is adapted from [29], (Paper B).

IDD (Real LR images)			
Method	NIMA↑	Meta-IQA↑	MOR↓
Bicubic [35]	4.73	0.330	-
ESRGAN [81]	4.94	0.325	-
MZSR [70]	5.00	0.330	2.96
DPSR [88]	4.92	0.330	3.16
RealSR [33]	4.83	0.296	4.88
DAN [48]	4.77	**0.330**	2.48
Ours	**5.03**	0.323	**1.45**

Table 2.4: *"Quantitative results on the IDD validation sets. ↑ and ↓ indicate whether higher or lower values are desired, respectively. As seen, our method obtains both the best MOR and NIMA results, and the second best Meta-IQA results."* [29]. This table is adapted from [29], (Paper B).

bers alone. Thus a visual comparison is also provided in Figures 2.7 and 2.8. As shown in Figure 2.7, SSG-RWSR produces a much cleaner image than the three models with the closest performance. While RealSR manages to recover a similar amount of detail the lines around the pharmacy cross are less sharp, and the texture is much less discernable. A similar trend is observed in Table 2.4 where the texture and sharpness produced by SSG-RWSR are much clearer compared to the competition. Noticeably, SSG-RWSR is also more robust to noise.

On Synthetic data, it can be observed that SSG-RWSR produces that are closest to the ground truth in terms of perceptual quality (i.e., LPIPS and DISTS) while achieving competitive performance according to traditional metrics (i.e., PSNR, and SSIM). When observing the examples shown in Figure 2.6 it can be seen that images produced by RealSR, and SSG-RWSR are sharper and can have higher frequency detail on textures. Whereas DPSR which scored the highest PSNR and SSIM is much similar to the HR ground-truth, it exhibits the characteristics of a blurry image, when compared to SSG-RWSR and RealSR.

Cityscapes (Synthesized LR images)				
Method	PSNR↑	SSIM↑	LPIPS↓	DISTS↓
Bicubic [35]	27.51	0.62	0.64	0.19
ESRGAN [81]	18.17	0.12	1.29	0.20
MZSR [70]	26.68	0.55	0.73	0.16
DPSR [88]	**33.11**	**0.90**	0.42	0.13
RealSR [33]	25.88	0.77	0.26	0.10
DAN [48]	27.16	0.58	0.60	0.20
Ours	29.08	0.83	**0.19**	**0.07**

Table 2.5: *"Quantitative results on the artificially degraded Cityscapes validation set. ↑ and ↓ indicate whether higher or lower values are desired, respectively. Our method achieves a good trade-off between low distortion and high perceptual quality with the second best PSNR and SSIM results, and the best perceptual quality as measured by the LPIPS and DISTS metrics."* [29]. This table is adapted from [29], (Paper B).

Furthermore, to see the impact of domain adaptation and semantic guidance, separate models were trained for each configuration. Shown in Table 2.5, the traditional domain adaptation approach in isolation performs significantly worse than semantic guidance in terms of perceptual quality, with SSG-RWSR further improving the performance. In the examples shown in Figure 2.9 it can be seen that the domain adaptation approach drives the model to reconstruct fine details, whereas the semantic guidance approach creates smoother and less noisy images. Finally, the combined approach manages to reconstruct finer detail while also producing a clearer image, effectively leveraging the benefits of both approaches.

Fig. 2.8: *"Comparison with SotA methods for ×4 SR of **real** images from the IDD dataset. As visible, our method reconstructs more detailed images with less artifacts compared to the existing methods."* [29]. This figure is adapted from [29], (Paper B).

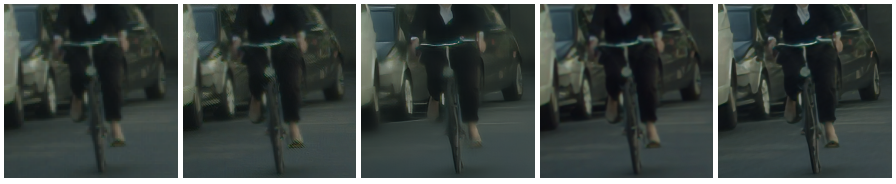


Fig. 2.9: *"Comparison with Bicubic interpolation, ESRGan [81] and various configurations of SSG-RWSR. Showing the impact of each module in the proposed SSG-RWSR framework. Ours$_{DA}$ denotes SSG-RWSR using only domain adaption, Ours$_{Guided}$ denotes SSG-RWSR using only semantic guidance, and finally Ours$_{Comb.}$ denotes the combination of domain adaptation and semantic guidance."* [29]. This figure is adapted from [29], (Paper B).

4.4 Summary and Contributions

In this section, the work conducted in Paper B was discussed. In large parts, the work addresses the problem of RWSR where ground truth LR-HR pairs are not available. To address this issue a novel framework (SSG-RWSR) was proposed, which leverages an auxiliary semantic segmentation network to guide the SR learning process. By doing so, the SR model can effectively adapt to the specific degradations present in real-world LR images, resulting in the reconstruction of images with sharp object boundaries and reduced noise. The contribution of the work described herein can be summarized as follows:

- The novel Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR) framework was proposed, which facilitates RWSR with guidance from an auxiliary SS task. With a focus on increasing perceptual consistency and removing noise.
- The proposed SSG-RWSR framework achieves superior results in terms of perceptual quality compared to SotA based on human evaluation.
- Domain adaptation and segmentation guidance are complimentary and help reconstruct textures and fine details, compared to domain-adaptation or segmentation guidance alone.

References

[1] A. Aakerberg, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Single-loss multi-task learning for improving semantic segmentation using super-resolution," in *Computer Analysis of Images and Patterns: 19th International Conference, CAIP 2021*. Springer, 2021, pp. 403–411.

[2] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR*, 2017.

[3] B. Baheti, S. Innani, S. Gajre, and S. N. Talbar, "Eff-unet: A novel architecture for semantic segmentation in unstructured environment," in *CVPR*, 2020.

[4] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2802–2819, 2018.

[5] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," in *NIPS*, 2019.

[6] J. Bruce, T. Balch, and M. Veloso, "Fast and inexpensive color image segmentation for interactive robots," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 3. IEEE, 2000, pp. 2061–2066.

[7] C.-F. R. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan, "Deep analysis of cnn-based spatio-temporal representations for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6165–6175.

[8] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[9] L. Chen, N. Ma, P. Wang, J. Li, P. Wang, G. Pang, and X. Shi, "Survey of pedestrian action recognition techniques for autonomous driving," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 458–470, 2020.

[10] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *ICCV*, 2017.

[11] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*. Springer, 2020, pp. 20–40.

[12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[13] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?" in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016, pp. 1–9.

[14] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?" in *WACV*, 2016.

[15] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2004.07728

[16] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *TPAMI*, 2016.

[17] H. T. Esfandarani and P. Milanfar, "NIMA: neural image assessment," *TIP*, 2018.

[18] M. Fritsche, S. Gu, and R. Timofte, "Frequency separation for real-world super-resolution," in *IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2019.

[19] H. Gajjar, S. Sanyal, and M. Shah, "A comprehensive study on lane detecting autonomous car using computer vision," *Expert Systems with Applications*, p. 120929, 2023.

[20] Y. Gao, M. Zhou, and D. N. Metaxas, "Utnet: a hybrid transformer architecture for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24*. Springer, 2021, pp. 61–71.

[21] R. Gerchberg, "Super-resolution through error energy reduction," *Optica Acta: International Journal of Optics*, vol. 21, no. 9, pp. 709–720, 1974.

[22] V. P. Goncalves, L. P. Silva, F. L. Nunes, J. E. Ferreira, and L. V. Araújo, "Concept drift adaptation in video surveillance: a systematic review," *Multimedia Tools and Applications*, pp. 1–41, 2023.

[23] M. Haris, G. Shakhnarovich, and N. Ukita, "Task-driven super resolution: Object detection in low-resolution images," in *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*. Springer, 2021, pp. 387–395.

[24] K. He, R. Girshick, and P. Dollár, "Rethinking imagenet pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4918–4927.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[26] F. Heuer, S. Mantowsky, S. Bukhari, and G. Schneider, "Multitask-centernet (mcn): Efficient and diverse multitask learning using an anchor free approach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 997–1005.

[27] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1439–1449.

[28] M. Huh, P. Agrawal, and A. A. Efros, "What makes imagenet good for transfer learning?" *arXiv preprint arXiv:1608.08614*, 2016.

[29] *Semantic segmentation guided real-world super-resolution*. IEEE, 2022.

[30] D. E. Ilea and P. F. Whelan, "Image segmentation based on the integration of colour–texture descriptors—a review," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2479–2501, 2011.

[31] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[32] A. Jha, A. Kumar, S. Pande, B. Banerjee, and S. Chaudhuri, "MT-UNET: A novel u-net based multi-task architecture for visual scene understanding," in *ICIP*, 2020.

[33] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, "Real-world super-resolution via kernel estimation and noise injection," in *CVPR Workshops*, 2020.

[34] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *CVPR-W*, 2016.

[35] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans Acoust. Speech Signal Process*, 1981.

[36] S. Kondo, "Lapformer: surgical tool detection in laparoscopic surgical video using transformer architecture," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2020.

[37] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, "Weighted ensemble object detection with optimized coefficients for remote sensing images," *ISPRS International Journal of Geo-Information*, vol. 9, no. 6, p. 370, 2020.

[38] 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.

[39] S. K. Kumaran, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *arXiv preprint arXiv:1901.08292*, 2019.

[40] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[41] C. Lennan, H. Nguyen, and D. Tran, "Image quality assessment," https://github.com/idealo/image-quality-assessment, 2018.

[42] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPR Workshops*, 2017.

[43] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1925–1934.

[44] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[45] L. Liu, S. Wang, and L. Wan, "Component semantic prior guided generative adversarial network for face super-resolution," *IEEE Access*, 2019.

[46] A. Lugmayr, M. Danelljan, and R. Timofte, "Unsupervised learning for real-world super-resolution," in *CVPR Workshops*.

[47] R. Lugmayr, M. Danelljan, and R. Timofte, "Ntire 2020 challenge on real-world image super-resolution: Methods and results," *CVPR Workshops*, 2020.

[48] Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan, "Unfolding the alternating optimization for blind super resolution," in *NeurIPS*, 2020.

[49] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, "Attentive single-tasking of multiple tasks," in *CVPR*, 2019.

[50] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[51] A. Milioto and C. Stachniss, "Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using cnns," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 7094–7100.

[52] A. Mishra, S. Kumar, T. Kalluri, G. Varma, A. Subramaian, M. Chandraker, and C. V. Jawahar, "Semantic segmentation datasets for resource constrained training," in *NCVPRIPG*, vol. 2, no. 3, 2020, p. 6.

[53] M. P. Mohammad Emad and H. Corporaal, "Dualsr: Zero-shot dual learning for real-world super-resolution," in *WACV*, 2021.

[54] F. Munir, S. Azam, and M. Jeon, "Sstn: Self-supervised domain adaptation thermal object detection for autonomous driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 206–213.

[55] Y. R. Musunuri and O.-S. Kwon, "Deep residual dense network for single image super-resolution," *Electronics*, vol. 10, no. 5, p. 555, 2021.

[56] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "Spotnet: Self-attention multi-task network for object detection," in *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 230–237.

[57] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4151–4160.

[58] D. Prasad, A. Gadpal, K. Kapadni, M. Visave, and K. Sultanpure, "Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 572–573.

[59] M. S. Rad, B. Bozorgtabar, U. Marti, M. Basler, H. K. Ekenel, and J. Thiran, "SROBB: targeted perceptual loss for single image super-resolution," in *ICCV*, 2019.

[60] M. S. Rad, B. Bozorgtabar, C. Musat, U. Marti, M. Basler, H. K. Ekenel, and J. Thiran, "Benefiting from multitask learning to improve single image super-resolution," *Neurocomputing*, 2020.

[61] N. C. Rakotonirina and A. Rasoanaivo, "Esrgan+: Further improving enhanced super-resolution generative adversarial network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3637–3641.

[62] V. Robert and H. Talbot, "Does super-resolution improve OCR performance in the real world? A case study on images of receipts," in *ICIP*, 2020.

[63] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *T-ITS*, vol. 19, no. 1, pp. 263–272, 2018.

[64] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICAI*, 2015, pp. 234–241.

[65] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[66] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[67] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *CVPR*, June 2018.

[68] A. A. Shvets, A. Rakhlin, A. A. Kalinin, and V. I. Iglovikov, "Automatic instrument segmentation in robot-assisted surgery using deep learning," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*. IEEE, 2018, pp. 624–628.

[69] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[70] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *CVPR*, 2020.

[71] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[72] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[73] A. Tao, K. Sapra, and B. Catanzaro, "Hierarchical multi-scale attention for semantic segmentation," *arXiv preprint arXiv:2005.10821*, 2020.

[74] Y. Tian and K. Bai, "End-to-end multitask learning with vision transformer," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[75] R. Timofte, E. Agustsson, L. Van Gool, M. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *CVPR Workshops*, 2017.

[76] G. Varma, A. Subramanian, A. M. Namboodiri, M. Chandraker, and C. V. Jawahar, "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments," in *WACV*, 2019.

[77] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2019.

[78] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[79] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in *CVPR*, 2020, pp. 3774–3783.

[80] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *CVPR*, 2018.

[81] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *ECCV*, 2019.

[82] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer vision and image understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[83] J. Xie, Y. Zheng, R. Du, W. Xiong, Y. Cao, Z. Ma, D. Cao, and J. Guo, "Deep learning-based computer vision for surveillance in its: Evaluation of state-of-the-art methods," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3027–3042, 2021.

[84] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *CVPR*, 2018.

[85] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," *arXiv preprint arXiv:1909.11065*, 2019.

[86] H. Z., J. S., X. Q., X. W., and J. J., "Pyramid scene parsing network," in *CVPR*, 2017.

[87] K. Zhang, L. Van Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3217–3226.

[88] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *CVPR*, 2019.

[89] Q. Zhang, G. Yang, and G. Zhang, "Collaborative network for super-resolution and semantic segmentation of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–12, 2021.

[90] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html

[91] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018.

[92] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2021.

[93] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *ECCV*, 2014.

[94] R. Zhou and S. Süsstrunk, "Kernel modeling super-resolution on real low-resolution images," in *ICCV*, 2019.

[95] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *CVPR*, 2020.

[96] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 2242–2251. [Online]. Available: https://doi.org/10.1109/ICCV.2017.244

[97] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *ICCV*, 2017.

Chapter 3

Paying Attention to Motion: Advancements in Video Transformers

1 Introduction

In the current landscape of computer vision, the proliferation of devices with cameras has enabled an unprecedented influx of visual data. From surveillance cameras capturing busy scenes of urban life to autonomous vehicles meticulously analyzing their surroundings to plan their next move. These camera sources yield continuous streams of visual data at an ever-increasing rate. When leveraging these continuous sources for inspection or observation, humans typically leverage temporal information to understand behavior, predict events, and provide context which can be employed to make informed decisions for a given point in time [47]. On the contrary, computer vision systems often process video streams one frame at a time [10, 27, 28, 77], which severely limits contextual information and forming temporal relationships, ultimately limiting understanding [13, 106]. Traditionally computer vision algorithms have relied on sequential processing techniques such as Recurrent Neural Networks (RNNs) and Long-Short-Term-Memorys (LSTMs) to capture temporal patterns and dependencies across frames [4, 47, 51]. Regrettably, these methods have been shown to struggle with modeling long-range dependencies [59], which is crucial for properly understanding intricate and complex relationships between distant frames.

Recent advancements in computer vision include the introduction of an attention-based architecture inspired by a recent advancement in Natural Language Processing (NLP), called transformers, or Vision Transformer (ViT) in

the computer vision domain. Transformers were initially introduced to address the challenge of modeling long-range dependencies, which can be crucial language tasks [12, 43, 94]. Their unique ability to process distant elements at the same time has shown promising results in traditional single-image computer vision tasks [30, 44, 61, 110], and have started to increase interest in the video domain [81].

The ability to leverage information across time to provide a contextual understanding of a given image could prove a crucial tool for accurate video understanding. From real-time action recognition and anomaly detection in surveillance videos to scene understanding and tracking in dynamic environments, the applications of Video Transformers (VTs) span a broad spectrum of real-world applications. Consequently, the combination of video data and transformers is interesting from an academic point of view but also presents novel methods that can be applied to industrial computer vision systems that process continuous streams of video data.

The rapid growth and popularity of Vision Transformers (ViTs) has proliferated across various computer vision tasks and has quickly taken the field by storm. Combined with their impressive ability to excel at various vision tasks, has made ViTs a promising avenue for real-world computer vision systems.

This chapter serves to provide insight into key components and challenges posed when applying transformers on video data, as well as discussion about current trends, methods and shortcomings commonly observed with Video Transformer (VT) namely:

Paper C **Key Challenges**: An overview of key challenges that transformers face when applied to video data. Particularly an overvew of challenges that persist from traditional transformers, as well as those unique to video.

Paper C **Trends of Video Transformers**: A broad overview of common trends adapted by VTs when handling video data. Particularly key trends regarding video transformers as a whole as well as those particularly aimed at object-centric tasks (i.e. Object detection, tracking, segmentation, etc.).

Paper C **Discussion and Implications**: Discussion addressing the efficacy and maturity of key trends as well as their implication for real-world deployment of computer vision system.

2 Background

In 2020, transformers were introduced to the visual domain, beating the performance of their CNN counterparts on fundamental vision tasks [7, 15]. However, requiring significantly more data to perform the pre-training step to beat

SotA by a significant margin. Albeit a data-hungry architecture, transformers quickly grew in popularity and in a short time have become a significant portion of publications in the academic field of computer vision [30, 44]. The exponential growth in publications has merited multiple surveys to establish an overview of ViTs [30, 44, 58, 61, 110, 113] and their ability to perform multimodal reasoning [84, 108], particularly language-vision learning [17, 66, 79].

Transformers' ability to learn long-range global relationships lends itself very well to data that contains a temporal component, such as video. However, the added dimension introduces computational cost due to the Self-Attention (SA) mechanism [94] as well as an abundance of redundant information [113].

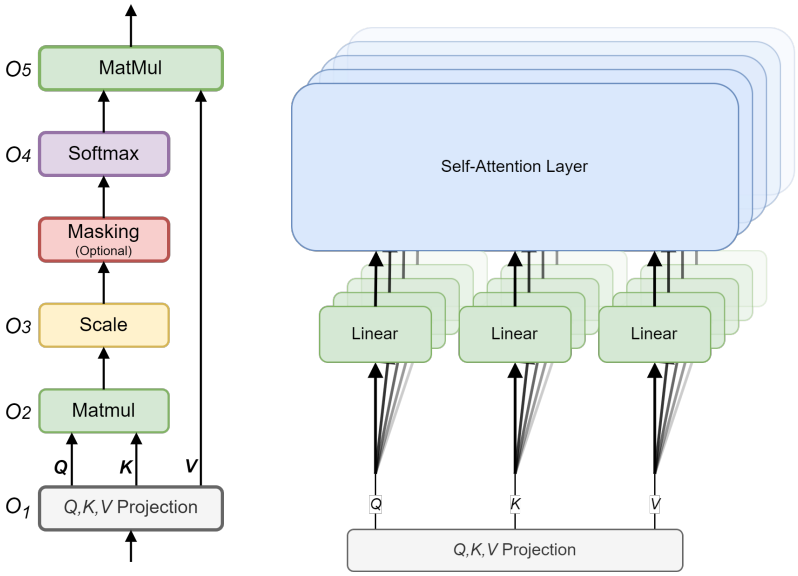
To properly understand the impacts and methods employed to address the temporal dimension of video, the fundamentals of the original transformer must be understood. In this section, a brief overview of the original transformer [94] is provided, with an accompanying description of its core component, i.e. SA. Subsequently, techniques necessary to adapt transformers from language to images are discussed.

2.1 The original transformer

Many NLP tasks are posed as a sequence-to-sequence problem, where a sequence of elements (i.e. sentence) is inputted and an output sequence is returned. Depending on the task the output would be different, e.g., In language translation; the equivalent sentence in the target language, in summarization; a sentence summarizing the input, etc. Traditional sequence-to-sequence models, such as LSTMs and RNNs, process information sequentially and capture dependencies between elements of an input sequence using a hidden state or memory.

In NLP, the input is typically a sequence of "tokens" (e.g., words). These tokens are an encoded representation of the word and typically correspond to a one-hot encoding of the word and its respective vocabulary [100]. While these encodings can be used directly, an embedding layer is typically used to project the vector to a continuous space, as it has been shown to improve the performance of some tasks [69]. The result is a sequence of continuous vector representations of the original input sequence. Finally, each token is augmented with a positional encoding which allows the network to discern the locality of a given token in the input sequence. The result is a sequence of token embeddings that can be processed by the transformer.

Similar to other SotA methods, transformers follow an encoder-decoder structure [116], where one module encodes information from the input sequence, which the decoder then employs to generate the appropriate output. Where the encoder only processes the input sequence once, the decoder works in an autoregressive manner, generating one output at a time and subsequently consuming the generated output, using it as an additional input. Both encoder



(a) Self-Attention **(b)** Multi-Head Self-Attention

Fig. 3.1: Visualization self-attention and multi-head self-attention proposed in [94]. This figure is adapted from [94]

and decoder modules consist of N transformer layers (visualized in Figure 3.2), with layers in the decoder, having an additional component allowing for cross-attending the representation generated by the encoder. Each transformer layer is composed of two components, namely a self-attention layer, and a position-wise feed-forward network. Additionally, each layer is wrapped with a residual connection and layer normalization [94]. To prevent the decoder module from attending beyond the token at the current position, a mask is applied so that the SA operation only attends to known outputs preceding the current position.

With an understanding of the input and the structure of the transformer, key elements of the transformer can be described. At a high level the SA mechanism (visualized in Figure 3.1a) can be broken into 5 steps:

O_1 **Query, Key & Value Pairs**: For each token in the input sequence three vectors are generated: a Query vector, a Key vector, and a Value vector. These vectors are obtained through a linear transformation and are used to learn the relationships between the input tokens.

O_2 **Similarity Scoring**: For each token in the input sequence a similarity score is calculated by computing the dot-product between Q at the current position, and K at all positions in the input sequence.

O_3 **Scaling**: To prevent the dot-products of O_2 from growing too large, they are counteracted with a scaling factor: $\frac{1}{\sqrt{dim_k}}$, where dim_k denotes the dimension of the K vectors.

O_4 **Attention Weighting**: The similarity scores are passed to a softmax function to obtain attention weights. These weights describe how much each token in the input sequence contributes to the output of the token at the current position.

O_5 **Weighted Summation**: the weighted sum of the V vectors and their attention weights results in a context vector for the current token. This context vector in essence contains information from all other tokens in the input sequence and emphasizes tokens wither more relevance (i.e. higher attention) to the current token.

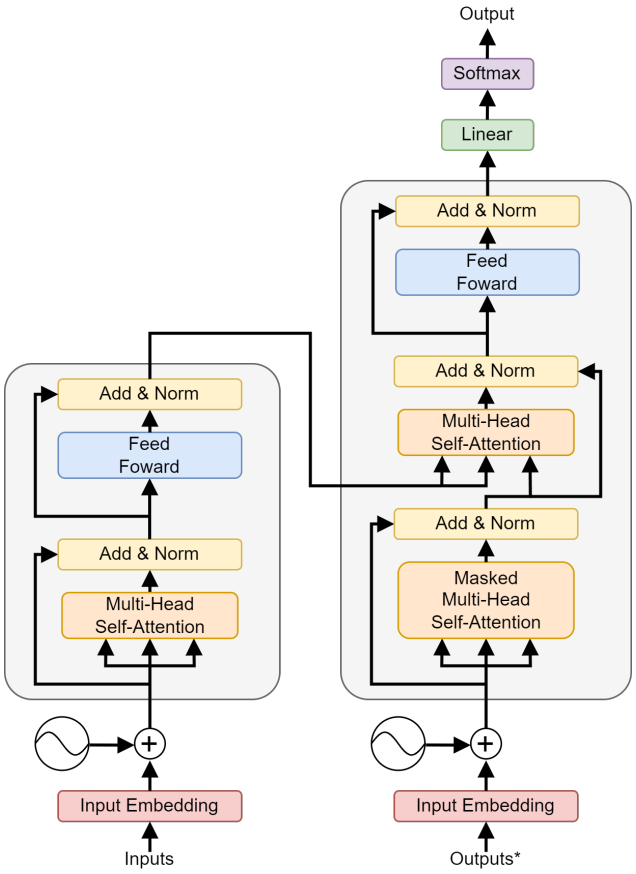


Fig. 3.2: Visualization of the original transformer proposed in [94]. This figure is adapted from [94]

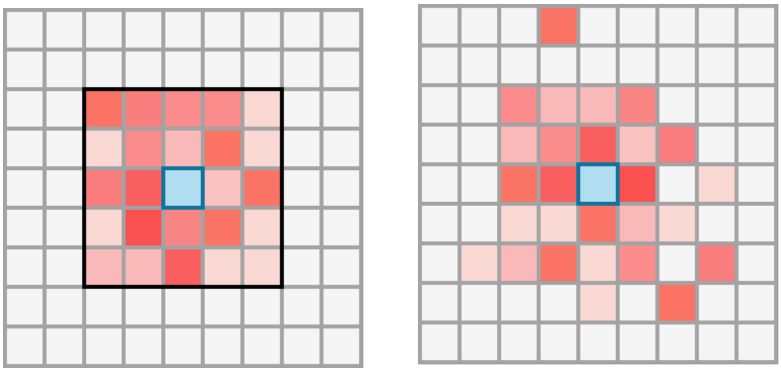
In their experiments, they found that instead of performing a single pass of self-attention, projecting the input sequence to several SA was beneficial. Each of these parallel heads has its own separate projection function (O_1), effectively allowing each pass of the SA step to attend to several things [94]. The implementation of this type of parallel SA is referred to as a Multi-Head Self-Attention (MHSA) layer.

Despite the improvements which are introduced with the attention style approach, it can be noted that the self-attention operation, particularly the pair-wise affinity calculation (O_2), scales quadratically with sequence length. Even though the self-attention operation is highly parallelizable, it can become computationally expensive for long input sequences.

2.2 Vision Transformers

When transitioning to the visual medium, the difference in the data structure is significant. Where natural language is structured as $K \times Tokens$ (K denoting the length of input tokens), images are typically structured as large tensors of shape: $H \times W \times C$ (denoting image Height, Width, and color Channels). Even images that otherwise are considered low-resolution would result in large sequences when un-rolled to fit the sequence structure expected by the transformer [15]. Early work addressed this by combining SA with CNNs to reduce the spatial dimensionality, and thus the sequence length [7, 97]. These architectures benefitted from the inductive neighborhood biases provided by the CNN and could leverage the global reasoning of transformers. Furthermore, it allowed large-scale unsupervised pre-training methods to be employed for training, reducing the reliance on annotated data [97].

However, in [78] they proposed that the SA style of attention could be used to replace traditional spatial convolutions, to achieve an attention-based model. Similarly to how the original transformer [94] allowed for more effective modeling of the global context, a fully-attention-based model could achieve the same for vision tasks. Extending on this work, [11] showed that full attention-based models like that proposed in [78], typically learned attention patterns similar to neighborhoods of convolutions and proved that theoretically, a single MHSA could express any convolution. These findings indicated that pure transformers could work for vision tasks, albeit restrictive due to the computational complexity of SA. With inspiration from these advances, a fully transformer-based approach was proposed, named ViT [15]. They proposed extracting $S \times S$ Non-overlapping patches from the image and using them as tokens. In accordance with the original transformer, these were then passed to a linear embedding function. Coincidentally, the embedding functions learned to extract patterns similarly to early filters in CNNs. ViT showed comparative performance with existing SotA methods and even surpassing SotA on certain tasks [15].



(a) Visualization of kernel weights of a convolution operation

(b) Visualization of attention weights of a SA operation

Fig. 3.3: Comparison between kernel weights and SA attention weights, showing the similarity between the receptive field of both operations. The blue square denotes the center of the kernel for convolutions and the queried token for SA. The intensity of the red color denotes the weight of the given pixel, higher saturation equates to a higher weight. On Figure 3.3a the limited window of the kernel is visualized as a black outline. Notably Figure 3.3a shows the inherent limitations of a convolutional operation, whereas Figure 3.3b shows that despite its global nature, the SA attention, can mimic that of a convolutional kernel.

2.3 Challenges for transformers in vision

In this section prominent challenges that apply to transformers in the vision domain are outlined, as VTs extend on the ideas and methods of ViTs, many challenges faced by ViTs also apply to VTs.

Explainibility: ViTs are still very young, and despite their popularity many, the explainability of the learned representations lacks detailed study. Most attempts to explain what elements in particular ViTs pay attention to too often revolve around visualizing the attention matrices of early transformer layers [7, 15, 31, 40]. Nonetheless, this could be a result of inspecting layers until an intuitive one is located. A deeper inspection of many architectures shows that you can obtain very similar attention maps from drastically different images [24, 76]. Recent work [24] has fed handcrafted images to ViTs to identify the types of patterns and relationships they learn. The study found that ViTs exhibits feature progression similar to that of CNNs, i.e. the level of abstraction increases with the depth of the model; from edges to object representations as progress is made towards the deeper layers [24, 40]. Furthermore, early embedding layers have been known to exhibit properties similar to early CNN filters, i.e. gradients and edges [15].

Fixed Sequence Length: Despite the SA working on arbitrary sequence length, the positional embedding layer does not allow for varied lengths of input sequences. Some works address this by interpolating between existing

positional embeddings [63, 109]. One of the solutions which have started to catch on is relative positional embeddings which could enable the use of variable length inputs [57, 65, 103]. However, most approaches that leverage relative positional encodings do not investigate this aspect

Data Consumption: Since ViTs do not inherently encode inductive biases (i.e. prior knowledge), they typically require a large corpus of data to identify fundamental patterns [11, 15, 44]. This is a potentially restrictive factor when adapting transformers to a new domain with little data. Research indicates that ViTs that have learned these patterns in one domain, translate very effectively to other similar domains [33, 67]. Recent ViTs have employed un-supervised pretext tasks to learn fundamental relationships in a domain thus the restrictive factor lies with the computational burden and data amount, as opposed to manual annotation [20, 32, 87]. Furthermore, There exists a distinct lack of research on the impact degraded, altered, or corrupted image data has on ViTs. As transformers rely on inter-token relationships, erroneous tokens could have unforeseen consequences for the SA operation.

Computational Complexity: As high fidelity cameras have become increasingly common [38], computer vision algorithms are expected to process larger HR images. With ViTs becoming a prominent architecture, their inability to scale effectively with larger inputs [121] stifles the adaptation of these algorithms. Additionally, research has shown that SA maps for a given token tend to favor specific neighbourhoods [11, 15, 64, 76], indicating that global-reasoning at all stages, might not be the ideal approach. Hierarchical approaches which progressively condense information display promising middle-ground [31, 64], by progressively expanding the receptive field of the SA follows the progressive modeling trend observed with the original ViT [24, 40]. Notably window-based attention approaches have linear complexity (with respect to window size) of the SA mechanism. Additionally, progressively condensing representations through repeated overlapping tokenization can facilitate a reduction in network depth and allow for larger input patches thus reducing input length [118].

Using a similar intuition, employing sparsity in the of the SA mechanism has also shown great promise [45, 102, 107, 125]. Using dynamically adjusted attention patterns [102, 107, 125] and dynamically masking SA to obtain a sparse attention mapping, can achieve comparative performance while reducing the computational burden. Alternatively, the length of the input sequence can be reduced by using a CNN as an embedding layer [50, 95, 125]. CNNs further allows the transformer to leverage the inductive biases of the CNN', which can facilitate the use of a shallow ViTs, further reducing cost [95].

3 Related Work

Transformers' popularity has only grown and proliferated throughout the NLP and computer vision fields since their inception. With the rapid advances in this field, a detailed overview is necessary to make informed decisions when aiming to employ transformers. Noting that at its core, VTs are adapted from ViTs, thus most challenges facing ViTs are transversal to VTs.

Existing surveys on transformers already cover: transformers in general [58], NLP [43], transformers in vision [61, 110], efficient designs [21, 90, 91], ViTs [30, 44, 113],, and multi-modality [79, 84, 108]. All though some surveys [30, 44, 79, 84, 108, 113] cover VTs, they do so superficially and contain insufficient detail regarding the challenges of modeling image sequences or the highly redundant spatiotemporal visuals present in videos.

3.1 Identifying Key Papers

To address the gap in the literature, a comprehensive analysis of works that leverage transformers to model video data, and their unique approaches and potentials was conducted in Paper C. To establish a structured overview of key insights an overview was established. The various trends and design choices observed in the VT pipeline, including input processing, architecture variations, and training methodologies were described in detail. Additionally, a detailed discussion of trends in object-centric and multimodal approaches is included. In this section the key insights as they relate to the thesis topic are outlined, for a detailed discussion and outline of other tasks as well as a comprehensive comparison of the self-reported performance of the video classification task see Paper C.

The timeframe for gathering papers started on the day the original transformer [94] was uploaded to arXiv.org and ended the window with the publication of CVPR2022 papers, marking the last set of publications from relevant venues prior to the submission of the survey, Paper C. This establishes the survey window as 12.06.2017 - 21.03.2022 (date format: DD.MM.YYYY).

To gather the relevant papers key terms were identified. These terms were aimed at identifying papers that leveraged transformer models and any other forms of self-attention (including non-local operations. The SA mechanism was considered to be the key element that defined transformers. However, the terminology had not fully solidified across topics, thus key search terms had to describe attention as a whole. Consequently, this resulted in an increase in false positives but also it likely prevented us from missing relevant papers.

For paper gathering the following high-impact venues were targeted: NeurIPS, TPAMI, CVPR, ICCV, ICLR, ICML, AIII, and ECCV. Papers containing a combination of 'Video' or 'Temporal' combined with any of the following

terms: transformer, attention, self-attention, non-local, and multi-head attention, were selected. Additionally, a manual search was conducted to identify key papers that were not published in high-impact venues. This resulted in 282 relevant papers, which were reviewed in detail. During the detailed review, a further 182 papers were excluded due to them not employing transformers to model video data, resulting in a collection of 100 key VT papers.

4 Insights and Trends of Video Transformers

Most advancements in VTs aim to handle the computational burden, often using frozen embedding networks (i.e., a pre-trained CNN) to reduce input dimensionality [29, 34, 42, 46, 49, 56, 60, 75]. However, this approach might limit transformers' ability to learn non-local low-level motion cues, which can be vital for a fine-grained understanding of dynamic scenes. Modeling temporal interactions in videos requires special considerations. Videos have highly redundant appearance information, which makes it challenging to create information-rich representations without repeating similar sub-representations [92, 122]. It has been shown that pure attentional models tend to lose expressiveness and exhibit uniform attention at deeper layers [14, 16, 39]. Many current VT designs and self-supervised learning approaches inherit from image approaches without considering temporal nuances, making them biased towards learning appearance features. Allowing temporal features to form at both low- and high-level while maintaining temporal fidelity is crucial. Thus, efforts to reduce redundancy in videos should primarily focus on appearance features. The introduction of novel VTs like MVITv2 [55] and SWINv2 [63] effectively address this with a progressive hierarchical approach that attends the temporal context prior to spatial aggregation. By favoring temporal attention prior to spatial attention, the architectures are able to better capture the fine-grained motion features.

4.1 Input-preprocessing Trends and Insights

While employing CNNs as an embedding network could limit the capabilities of transformers, inductive biases have proven to be hugely beneficial with regard to performance [26, 35, 63, 72]. However, there is a distinct lack of novel VT designs, like MViT [19] which leverage inductive biases embedded directly into the transformer, despite being able to achieve SotA performance while retaining comparable complexity to CNNs counterparts. Alternatively, Some approaches attempt to infuse inductive biases into the transformer through external networks, such as object detectors [26, 35], 3D action features [124], or localized spatial features [22], essentially providing the transformer with task-specific priors. A few novel approaches [35, 72] employ motion-informed

tokens to induce motion-specific biases. Combined with the novel architectures such as MVIT [36] and SWINv2 [63], motion biases could be leveraged to refine predictions in challenging scenarios with low-confidence appearance-based features (e.g., visibility is severely reduced or obscured).

4.2 Architectural Trends and Insights

Attempts at reducing the computational burden of pure transformer models typically leverage approaches that restrict or limit the self-attention mechanism to a specific neighborhood [3, 6, 29, 63, 65, 70, 103, 112], axis [2, 3, 18, 53, 93, 119] or pattern [3, 53, 72, 120]. Limiting the scope of the SA mechanism consequently reduces the type of relationships that can be modeled. Particularly self-attention targeted towards a specific neighborhood will enforce CNN like patterns of attention, thus requiring an additional mechanism to obtain information from the global context. To address this some works [36, 55, 64, 65] leverage a progressive refinement where the receptive field of the restricted attention is over-lapped or shifted progressively to obtain a global context at deeper layers. This enforces a similar pattern to those observed in CNNs and large-scale ViTs, i.e., increasingly higher levels of abstraction and receptive field of the attention mechanism with each layer [11]. This would indicate that early layers of transformers could be limited to a local neighborhood (like convolutions) to reduce computational complexity at earlier layers, however, it could be argued that this restriction would be somewhat equivalent to leveraging CNN backbones.

Similar to ViTs, research on VTs struggles with the explainability of the produced representations. While there is a high level of understanding that later layers target broader concepts rather than local patterns, the understanding of why certain local and non-local patterns are preferred at different stages is still not fully understood. Current approaches tend to overlay heatmaps of specific attention heads as an ad-hoc explanation to the learned relationships [5, 26, 39, 41, 52, 70, 71, 73, 74, 82, 105], commonly highlighting heatmaps that are perceptually intuitive to a human observer [105]. Attempts at providing explainability through visualizing attention typically also only address spatial attention. Visualizing attention further is very unintuitive when applied to video sequences, as it would require per-frame inspection of attention from a specific token of a specific frame. Even then it does not provide much insight into the spatio-temporal attention patterns. Understanding these attention patterns could provide invaluable insights into relevant design choices, but is currently severely understudied.

4.3 Training Methodology Trends and Insights

Similar to the use of pre-trained CNN backbone networks, many VT approaches directly adopt ViT architectures, and thus are able to initialize from their ViT counterpart [2, 6, 26, 70, 111, 114]. While initializing from a pre-trained transformer typically reduces the required training time significantly, it initiates the VT in a state heavily favoring spatial relationships, which might result in a reliance on appearance-based features rather than motion-based features. Some VTs [96, 101], attempt to avoid this by initializing from a model pretrained on video classification.

While transformers can be trained end-to-end on down-stream tasks [2, 3, 19, 39, 103, 120] like traditional CNNs, a large component of what makes transformers unique, is their reliance on inter-token relationships. In combination with the input inherently being separated into segments (i.e. tokens) lends itself well to self-supervised pre-training, particularly Masked-Token-Modeling (MTM). MTM takes inspiration from BERT [12], where a subset of tokens are randomly replaced by a learnable $[MSK]$ token which the network is tasked with predicting. Due to the continuous nature of images, it would require mapping to 255^3 distinct elements. To address this, common MTM approaches leverage feature-level regression [8, 52, 54, 101] or contrastive [9, 48, 52, 85] approaches to guide the network reconstruction. While uncommon, some work has managed to regress pixel-level tokens directly [63, 92]. Notably, VMAE [32, 92] have shown that it is possible to reconstruct complete frames and partial video sequences with remarkable fidelity, even with large masking ratios (e.g., 90%). This further underlines the redundancy present in the video domain. Self-supervised pretraining generally shows great promise for generalization, as it provides a strong avenue for learning from very large corpora of data, thus learning robust relationships that generalize well. While only a few have studied out-of-distribution data [60, 62, 73, 83, 104, 124] or cross-domain evaluation [25, 86, 89, 117], performance of VTs remains rather consistent when tested. Thus, VTs are potentially strong candidates for real-world vision applications. Real-world camera systems could introduce different or varied sampling rates than that present in the training data. The impact of changing temporal resolution in such a manner is still not investigated and remains an open-ended question.

4.4 Multi-Modal Insights

Video is inherently multi-modal in that it contains both visual and auditory information. Previous work has shown that the high-level semantic representations learned by transformers transfer well to other modalities [67, 88]. In combination with transformers' lack of inductive biases, transformers may enable them to learn shared multi-modal representation spaces, leading to better

generalization capabilities. A few works have shown to align representations from multiple modalities via instance-based modeling [25, 48, 52, 85]. For example, VATT [1] is able to perform heavy downsampling of video by aligning it with audio and textual modalities. STiCA [74] learn to attend to spatial sources of audio within the video by aligning audio with visual crops. In some cases, it has even been shown that sharing weights between transformers can improve alignment as well as the performance of downstream tasks [1, 48]. This kind of alignment has previously been shown to be useful for video classification with CNNs [80].

4.5 Object-Centric Insights

Advances regarding object-centric tasks like object detection, tracking, and segmentation are particularly interesting. As they are typically focused on per-object outputs, redundant information is present both temporally, as well as within a given frame. As such, recent works [35, 37, 68, 112, 123] have adopted producing object-centric tokens which can be used to correlate specific object instances temporally. Notably, some approaches [35, 68] leverage a memory buffer of object representations which can then auto-regressively aggregate information between frames. These 'messenger' tokens can be used to distill relevant information between frames, effectively reducing the number of tokens attended by the transformer, making it more computationally effective. The IFC-transformer [37], in particular, processes these object tokens in an isolated encoder, thus learning inter-object relationships completely separated from the context. To obtain contextually appropriate relationships, the transformer has to encode the relevant information from the entire image in the object token. GroupFormer [53] leverages this principle to perform action classification of individuals in a scene through an auxiliary branch. These refined object representations can then be leveraged to reason group behavior, thus allowing for scene understanding of individual objects and emergent group behavior simultaneously. Alternatively, TeViT [112] progressively shifts these object tokens to adjacent frames and encodes information from multiple frames at a time. The benefit of this progressive encoding is that the object representation is an aggregate from multiple frames rather than a gradual refinement of a specific frame. Another method [114] leverages both short-term and long-term information sharing by leveraging object representations from distant and close frames in parallel and shows that these methods can help produce smooth and continuous object detections and tracking, even of heavily occluded objects.

Similarly for the segmentation task, recent work [98, 99, 115, 117] leverage temporal representations to refine intermediate representations steps. Notably, [99] leverages an auxiliary matching loss, which temporally matches representations to implicitly learn to track. This allows the model to poten-

tially draw information about texture from multiple time steps and could potentially help discern motion-induced degradation such as motion blur.

4.6 Implications for Real-World Applications

The field of VTs is still rapidly evolving and trends have emerged, both for general applications as well as task-specific applications. The choice of VT for real-world applications will depend strongly on the available compute and the granularity of the downstream tasks. One of the troublesome trends is the widespread use of CNNs as embedding networks [29, 34, 42, 46, 49, 56, 60, 75], as it relies on the chosen networks' ability to effectively extract patterns. Intuitively this would mean that challenges such as concept drift and out-of-distribution data, would be difficult to overcome when relying on an embedding network that is ill-suited for the task. Novel architectures such as SWIN and MVIT [36] seem more promising methods due to inducing neighborhood inductive biases through a hierarchical approach which learn representations similar to that of fully-attentional models but with much lower computational complexity [55, 63, 64].

One of the more novel avenues that transformers enable is the ability to encode multi-modal representations and object-/context-specific information across time [68, 98, 99, 114]. While they have typically employed representations adjacent to their specific task (e.g., bounding boxes and visual features for object detection), it could be generalized to leverage contextual clues from auxiliary systems or sensors which can guide/assist the primary modality. Consequently, VTs display traits which could have promising impacts on multi-sensory video systems.

Coupled with the advances in efficient design, the field is rapidly approaching a state where the widespread use of transformer-style models can be applied in real-world applications. Their ability to employ non-local reasoning (both temporally and across modalities) makes them very interesting candidates for fine-grained analysis of video data. However, to see deployments of such systems outside of specialized use cases is still limited by the computational complexity of these large models. While they are approaching comparable computational complexity with SotA CNNs, most real-world deployments still leverage very light-weight CNNs [10, 23, 38].

5 Summary and Contributions

The introduction of transformers in NLP and their subsequent adaptation to the visual domain brought significant advancements in both fields. transformers' attention-based architecture allowed for global context understanding, outperforming traditional recurrent networks in language tasks and CNNs in

computer vision tasks. However, applying transformers to video data introduced challenges such as computational cost, redundant information, and the need for effective temporal modeling. Despite some existing surveys, a comprehensive analysis of how to effectively employ transformers on video data was lacking. Therefore, in Paper C we conducted a literature review on video transformers and found 282 relevant papers from which 100 key papers were identified and analyzed. The main contributions within the field of Video Transformer (VT) are thus:

- A systematic review of the field of VTs, documenting trends and patterns of existing work. Identifying key papers and novel architectures which show great promise for adapting transformers to video.
- A detailed review of existing methods and approaches to deal with the large dimensionality of videos. Providing insights into the trends and patterns related to the adaptation of ViTs to video data.
- Highlighted the need for future research on the explainability and consequences of certain design choices, to achieve a more detailed understanding of the spatial and temporal relationships that are modeled by the transformer.
- A discussion of the potential impact of video-specific design choices in terms of the learning capabilities of the transformer and potential implications for deployment on real-world data.

Overall, this section provides a comprehensive overview of the main challenges and advancements in adapting transformers to video data, offering valuable insights and directions for future research.

References

[1] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," *NeurIPS*, 2021.

[2] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021.

[3] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, 2021.

[4] P. V. K. Borges, N. Conci, and A. Cavallaro, "Video-based human behavior understanding: A survey," *IEEE transactions on circuits and systems for video technology*, vol. 23, no. 11, pp. 1993–2008, 2013.

[5] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner, "Transformerfusion: Monocular rgb scene reconstruction using transformers," *NeurIPS*, 2021.

[6] A. Bulat, J. M. Perez Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos, "Space-time mixing attention for video transformer," *NeurIPS*, 2021.

[7] 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: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[8] J. Chen and H. Chao, "Videotrm: Pre-training for video captioning challenge 2020," in *ACM-MM*, 2020.

[9] X. Chen, D. Liu, C. Lei, R. Li, Z.-J. Zha, and Z. Xiong, "Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer," in *ACM-MM*, 2019.

[10] S. S. Chouhan, U. P. Singh, and S. Jain, "Applications of computer vision in plant pathology: a survey," *Archives of computational methods in engineering*, vol. 27, pp. 611–632, 2020.

[11] J.-B. Cordonnier, A. Loukas, and M. Jaggi, "On the relationship between self-attention and convolutional layers," in *ICLR*, 2019.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Computational Linguistics*, 2019.

[13] A. Diba, M. Fayyaz, V. Sharma, M. Paluri, J. Gall, R. Stiefelhagen, and L. Van Gool, "Large scale holistic video understanding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*. Springer, 2020, pp. 593–610.

[14] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2793–2803.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[16] 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," in *ICLR*, 2021.

[17] Y. Du, Z. Liu, J. Li, and W. X. Zhao, "A survey of vision-language pre-trained models," in *IJCAI*, 2022.

[18] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "Sstvos: Sparse spatiotemporal transformers for video object segmentation," in *CVPR*, 2021.

[19] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *ICCV*, 2021.

[20] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, "A large-scale study on unsupervised spatiotemporal representation learning," in *CVPR*, 2021.

[21] Q. Fournier, G. M. Caron, and D. Aloise, "A practical survey on faster and lighter transformers," *arXiv*, 2021.

[22] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 214–229.

[23] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine vision and applications*, vol. 25, pp. 245–262, 2014.

[24] A. Ghiasi, H. Kazemi, E. Borgnia, S. Reich, M. Shu, M. Goldblum, A. G. Wilson, and T. Goldstein, "What do vision transformers learn? a visual exploration," *arXiv preprint arXiv:2212.06727*, 2022.

[25] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox, "Coot: Cooperative hierarchical transformer for video-text representation learning," in *NeurIPS*, 2020.

[26] R. Girdhar and K. Grauman, "Anticipative video transformer," in *ICCV*, 2021.

[27] J. F. S. Gomes and F. R. Leta, "Applications of computer vision techniques in the agriculture and food industry: a review," *European Food Research and Technology*, vol. 235, pp. 989–1000, 2012.

[28] A. Gowen, B. Tiwari, P. Cullen, K. McDonnell, and C. O'Donnell, "Applications of thermal imaging in food quality and safety assessment," *Trends in food science & technology*, vol. 21, no. 4, pp. 190–200, 2010.

[29] Y. Gu, L. Wang, Z. Wang, Y. Liu, M.-M. Cheng, and S.-P. Lu, "Pyramid constrained self-attention network for fast video salient object detection," in *AAAI*, 2020.

[30] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," in *IEEE TPAMI*, 2022.

[31] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.

[32] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.

[33] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve out-of-distribution robustness," in *ACL*, 2020.

[34] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake detection scheme based on vision transformer and distillation," *arXiv*, 2021.

[35] R. Herzig, E. Ben-Avraham, K. Mangalam, A. Bar, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson, "Object-region video transformers," in *CVPR*, 2022.

[36] F. Heuer, S. Mantowsky, S. Bukhari, and G. Schneider, "Multitask-centernet (mcn): Efficient and diverse multitask learning using an anchor free approach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 997–1005.

[37] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim, "Video instance segmentation using inter-frame communication transformers," *NeurIPS*, 2021.

[38] M. Intelligence, "Ir camera market - growth, trends, covid-19 impact, and forecasts (2021 - 2026)," https://www.mordorintelligence.com/industry-reports/ir-camera-market, 2021, accessed: 2021-08-11.

[39] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *ICML*, 2021.

[40] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[41] ——, "Attention is not explanation," in *NAACL-HLT*, 2019.

[42] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *CVPR*, 2020.

[43] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS : A survey of transformer-based pretrained models in natural language processing," *arXiv*, 2021.

[44] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM CSUR*, 2022.

[45] K. Kim, B. Wu, X. Dai, P. Zhang, Z. Yan, P. Vajda, and S. J. Kim, "Rethinking the self-attention in vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3071–3075.

[46] S. Kondo, "Lapformer: surgical tool detection in laparoscopic surgical video using transformer architecture," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2020.

[47] G. Lavee, E. Rivlin, and M. Rudzsky, "Understanding video events: A survey of methods for automatic interpretation of semantic occurrences in video," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 5, pp. 489–504, 2009.

[48] S. Lee, Y. Yu, G. Kim, T. Breuel, J. Kautz, and Y. Song, "Parameter efficient multimodal transformers for video representation learning," in *ICLR*, 2021.

[49] J. Lei, L. Wang, Y. Shen, D. Yu, T. L. Berg, and M. Bansal, "Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning," in *ACL*, 2020.

[50] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 619–13 627.

[51] F. Li, C. Gan, X. Liu, Y. Bian, X. Long, Y. Li, Z. Li, J. Zhou, and S. Wen, "Temporal modeling approaches for large-scale youtube-8m video understanding," *arXiv preprint arXiv:1707.04555*, 2017.

[52] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," in *EMNLP*, 2020.

[53] S. Li, Q. Cao, L. Liu, K. Yang, S. Liu, J. Hou, and S. Yi, "Groupformer: Group activity recognition with clustered spatial-temporal transformer," in *ICCV*, 2021.

[54] S. Li, X. Li, J. Lu, and J. Zhou, "Self-supervised video hashing via bidirectional transformers," in *CVPR*, 2021.

[55] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *CVPR*, 2022.

[56] Z. Li, Z. Li, J. Zhang, Y. Feng, and J. Zhou, "Bridging text and video: A universal multimodal transformer for audio-visual scene-aware dialog," *IEEE/ACM TASLP*, 2021.

[57] J. Lin and S.-h. Zhong, "Bi-directional self-attention with relative positional encoding for video summarization," in *ICTAI*, 2020.

[58] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[59] D. Linsley, J. Kim, V. Veerabadran, C. Windolf, and T. Serre, "Learning long-range spatial dependencies with horizontal gated recurrent units," *Advances in neural information processing systems*, vol. 31, 2018.

[60] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *ICCV*, 2021.

[61] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, "A Survey of Visual Transformers," *arXiv*, 2021.

[62] Y.-T. Liu, Y.-J. Li, F.-E. Yang, S.-F. Chen, and Y.-C. F. Wang, "Learning hierarchical self-attention for video summarization," in *ICIP*, 2019.

[63] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *CVPR*, 2022.

[64] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[65] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *CVPR*, 2022.

[66] S. Long, F. Cao, S. C. Han, and H. Yang, "Vision-and-language pretrained models: A survey," in *IJCAI*, 2022.

[67] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Frozen pretrained transformers as universal computation engines," *AAAI CAI*, 2022.

[68] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *CVPR*, 2022.

[69] B. Mitra and N. Craswell, "Neural text embeddings for information retrieval," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 2017, pp. 813–814.

[70] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *ICCV*, 2021.

[71] A. Pashevich, C. Schmid, and C. Sun, "Episodic transformer for vision-and-language navigation," in *ICCV*, 2021.

[72] M. Patrick, D. Campbell, Y. Asano, I. Misra, F. Metze, C. Feichtenhofer, A. Vedaldi, and J. F. Henriques, "Keeping your eye on the ball: Trajectory attention in video transformers," *NeurIPS*, 2021.

[73] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. G. Hauptmann, J. F. Henriques, and A. Vedaldi, "Support-set bottlenecks for video-text representation learning," in *ICLR*, 2021.

[74] M. Patrick, P.-Y. Huang, I. Misra, F. Metze, A. Vedaldi, Y. M. Asano, and J. a. F. Henriques, "Space-time crop & attend: Improving cross-modal video representation learning," in *ICCV*, 2021.

[75] D. Purwanto, R. Renanda Adhi Pramono, Y.-T. Chen, and W.-H. Fang, "Extreme low resolution action recognition with spatial-temporal multi-head self-attention and knowledge distillation," in *CVPR*, 2019.

[76] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[77] M. Rai, T. Maity, and R. Yadav, "Thermal imaging system and its real time applications: a survey," *Journal of Engineering Technology*, vol. 6, no. 2, pp. 290–303, 2017.

[78] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *NeurIPS*, 2019.

[79] L. Ruan and Q. Jin, "Survey: Transformer based video-language pre-training," *AI Open*, 2022.

[80] M. C. Schiappa, Y. S. Rawat, and M. Shah, "Self-supervised learning for videos: A survey," *arXiv*, 2022.

[81] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés, "Video transformers: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[82] S. Serrano and N. A. Smith, "Is attention interpretable?" in *ACL*, 2019.

[83] J. Shao, X. Wen, B. Zhao, and X. Xue, "Temporal context aggregation for video retrieval with contrastive learning," in *WACV*, 2021.

[84] A. Shin, M. Ishii, and T. Narihira, "Perspectives and prospects on transformer architecture for cross-modal tasks with language and vision," *IJCV*, 2022.

[85] C. Sun, F. Baradel, K. Murphy, and C. Schmid, "Contrastive bidirectional transformer for temporal representation learning," *arXiv*, 2019.

[86] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *ICCV*, 2019.

[87] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[88] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *CVPR*, 2022.

[89] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," in *ICCV*, 2021.

[90] Y. Tay, M. Dehghani, S. Abnar, Y. Shen, D. Bahri, P. Pham, J. Rao, L. Yang, S. Ruder, and D. Metzler, "Long range arena: A benchmark for efficient transformers," *arXiv*, 2020.

[91] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM CSUR*, 2020.

[92] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *NeurIPS*, 2022.

[93] T.-D. Truong, Q.-H. Bui, C. N. Duong, H.-S. Seo, S. L. Phung, X. Li, and K. Luu, "Direcformer: A directed attention in transformer approach to robust action recognition," in *CVPR*, 2022.

[94] 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 *NeurIPS*, 2017.

[95] C. Wang, H. Xu, X. Zhang, L. Wang, Z. Zheng, and H. Liu, "Convolutional embedding makes hierarchical vision transformer stronger," in *European Conference on Computer Vision*. Springer, 2022, pp. 739–756.

[96] J. Wang, G. Bertasius, D. Tran, and L. Torresani, "Long-short temporal contrastive learning of video transformers," in *CVPR*, 2022.

[97] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018.

[98] Y. Wang, Z. Liu, Y. Xia, C. Zhu, and D. Zhao, "Spatiotemporal module for video saliency prediction based on self-attention," *Image and Vision Computing*, 2021.

[99] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *CVPR*, 2021.

[100] J. J. Webster and C. Kit, "Tokenization as the initial phase in nlp," in *COLING 1992 volume 4: The 14th international conference on computational linguistics*, 1992.

[101] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *CVPR*, 2022.

[102] C. Wei, B. Duke, R. Jiang, P. Aarabi, G. W. Taylor, and F. Shkurti, "Sparsifiner: Learning sparse instance-dependent attention for efficient vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 22 680–22 689.

[103] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," in *ICLR*, 2020.

[104] W. Weng, Y. Zhang, and Z. Xiong, "Event-based video reconstruction using transformer," in *ICCV*, 2021.

[105] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *EMNLP-IJCNLP*, 2019.

[106] C.-Y. Wu and P. Krahenbuhl, "Towards long-form video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1884–1894.

[107] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803.

[108] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: a survey," *arXiv*, 2022.

[109] R. Xu, X. Wang, K. Chen, B. Zhou, and C. C. Loy, "Positional encoding as spatial inductive bias in gans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 569–13 578.

[110] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, "Transformers in computational visual media: A survey," *Computational Visual Media*, 2022.

[111] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid, "Multiview transformers for video recognition," in *CVPR*, 2022.

[112] S. Yang, X. Wang, Y. Li, Y. Fang, J. Fang, W. Liu, X. Zhao, and Y. Shan, "Temporally efficient vision transformer for video instance segmentation," in *CVPR*, 2022.

[113] Y. Yang, L. Jiao, X. Liu, F. Liu, S. Yang, Z. Feng, and X. Tang, "Transformers meet visual learning understanding: A comprehensive review," *arXiv*, 2022.

[114] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," *NeurIPS*, 2021.

[115] L. Ye, M. Rochan, Z. Liu, X. Zhang, and Y. Wang, "Referring segmentation in images and videos with cross-modal self-attention network," *TPAMI*, 2021.

[116] H. Yousuf, M. Lahzi, S. A. Salloum, and K. Shaalan, "A systematic review on sequence-to-sequence learning with neural network and its models." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 11, no. 3, 2021.

[117] B. Yu, W. Li, X. Li, J. Lu, and J. Zhou, "Frequency-aware spatiotemporal transformers for video inpainting detection," in *ICCV*, 2021.

[118] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 558–567.

[119] S. Yun, J. Kim, D. Han, H. Song, J.-W. Ha, and J. Shin, "Time is MattEr: Temporal self-supervision for video transformers," in *ICML*, 2022.

[120] X. Zha, W. Zhu, L. Xun, S. Yang, and J. Liu, "Shifted chunk transformer for spatio-temporal representational learning," *NeurIPS*, 2021.

[121] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 104–12 113.

[122] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE TPAMI*, 2012.

[123] S. Zheng, S. Chen, and Q. Jin, "Vrdformer: End-to-end video visual relation detection with transformers," in *CVPR*, 2022.

[124] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *CVPR*, 2020.

[125] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

Chapter 4

Rising Temperatures: Exploring Thermal Concept Drift

1 Introduction

In recent years, the increased use of computer vision in deployed systems has helped transform numerous industries and domains, offering unprecedented capabilities and opportunities in terms of automated image analysis [61]. Supported by the advancements of specialized processors and GPUs, powerful deep-learning models trained on large-scale annotated datasets, have become increasingly common. With this, computer vision has become an integral component in many vision applications [21, 46, 49]. For autonomous vehicles, computer vision enables real-time detection, recognition, and tracking, allowing vehicles to perceive their surroundings and make informed decisions [42, 98]. In retail, vision-based systems have facilitated the implementation of automated checkout processes, inventory monitoring, and behavior analysis, leading to improved efficiency, reduced waste, and an improved customer experience. Moreover, computer vision has become an invaluable tool in detecting and analyzing various security scenarios. In surveillance this has allowed for automated monitoring and analysis of video footage, alleviating the burden of human operators and enhancing overall efficiency. Overall, the increased utilization of vision algorithms in deployed systems is opening up new possibilities and driving innovation across numerous industries [21, 42, 46, 49, 116, 132, 145].

For contexts involving adverse weather conditions, the integration of thermography becomes ideal, due to its inherent advantages over traditional image

cameras. Thermography, also known as thermal imaging, excels at image acquisition with minimal impact from adverse weather and lighting conditions, due to capturing Infrared Radiation (IR) instead of visible light. Consequently, this makes thermal cameras an ideal deployment of computer vision algorithms in contexts where continuous operation is expected [36, 41, 55, 131]. There exist various types of thermal cameras aimed at solving a range of different problems. Within thermography, there are two distinct types of cameras. The first employs qualitative thermography, also known as relative thermal imaging, which seeks to portray the relative differences in IR throughout the camera's Field-of-View (FoV). Primarily employed for inspection and security applications, qualitative thermography excels at providing discernible contrasts between colder and hotter elements within the scene, irrespective of absolute temperature values. The second employs quantitative thermography, or absolute thermal imaging, which meticulously maps each sampling point in the camera's field of view to an absolute temperature measurement, enabling a precise assessment of thermal discrepancies between distinct elements present in the scene. The benefit of quantitative thermography lies in its consistent visual response to any thermal signature encountered, facilitating accurate temperature assessments with superior resolution and reliability [131]. Absolute thermal cameras have all the capabilities of relative thermal cameras with a precise mapping between response and absolute temperature, which is a compelling argument to leverage this technology. However, the construction of absolute thermal cameras is significantly more intricate compared to relative thermal cameras, thus making it more costly to produce [40]. Consequently, most consumer applications leverage qualitative thermography to reduce costs [61].

Deploying algorithms to real-world contexts further present several challenges such as domain shifts, perspectives, and object configurations which are typically not observed in the training data [5]. To alleviate some of these problems large-scale datasets with a lot of diversity and variation are often used for training or pretraining, to allow the algorithm to generalize better. Nonetheless, the algorithm cannot be expected to have seen the diversity and variability that will be observed during deployment. Furthermore, in a real-world context, these challenges are further exacerbated by the algorithm potentially being deployed on several different input sources [61]. Having varying input sources potentially introduces visual changes, in terms of image resolution, image quality, compression artifacts, etc. Consequently, degradation of performance is often expected during deployment.

In academia, the validation of a computer vision system is traditionally focused on training and evaluating the system on public large-scale datasets [24, 48, 59]. Following this methodology allows for benchmarking and comparing algorithms with existing methods, providing insights into their efficacy. Large-scale datasets are typically used to enable a thorough evaluation of the

algorithm's ability to generalize as well as performance on unseen data. However, it is important to acknowledge that this type of algorithm validation does not necessarily translate to real-world performance. Notably, public large-scale thermal datasets commonly used in academia are captured using expensive absolute thermal cameras [24, 59] making them ill-suited for evaluation and training of models deployed on relative-thermal camera systems [131].

Consequently, evaluating the impact of concept drift during the long-term deployment of computer vision algorithms poses a uniquely difficult challenge. To bridge the gap between research and the real-world application of generic object-detection models, identifying parameters that induce concept drift and evaluating with respect to the identified parameters is a crucial step [5, 89].

This chapter serves to provide insight into three problems related to the long-term deployment of thermal computer vision systems, namely:

Paper D **Thermal Concept Drift during Long-term Deployment**: An examination of techniques to identify concept drift specifically in thermal imagery with associated weather data, aiming to capture the evolving nature of thermal object detection over extended periods. In addition to providing a large-scale dataset for comparison and evaluation with extensive metadata.

Paper E **Evaluating Long-term Robustness under Concept Drift**: An investigation and challenge posed at ECCV2023 for evaluating algorithm resilience when trained on limited data and evaluated on long-term data.

Paper F **Training Weather-aware Detection Algorithms**: A study of different conditioning methods for CNNs and Transformers, using weather-related data to guide optimization.

2 Background

The core objective of computer vision algorithms is to solve tasks based on visual input, which requires identifying and recognizing patterns, structures, and relationships from visual inputs. These patterns can range from simple geometric shapes to complex objects. These patterns can be tied to an optimization goal, such as object detection, image classification, action recognition, scene understanding, etc. To achieve this goal, the network must gain an understanding of the patterns and relationships that make up a certain concept. When the system is exposed to an unseen data distribution due to changes in the observed context, these concepts can change resulting in undesired performance impacts.

2.1 Concept Drift

The term concept drift describes the underlying shift in data distribution and thereby the recognized patterns associated with a given context. However, the pattern in which it manifests can vary greatly depending on the origin of the induced drift. Despite real-world concept drift often being a combination of multiple drift types, drift is typically categorized into one of four categories:

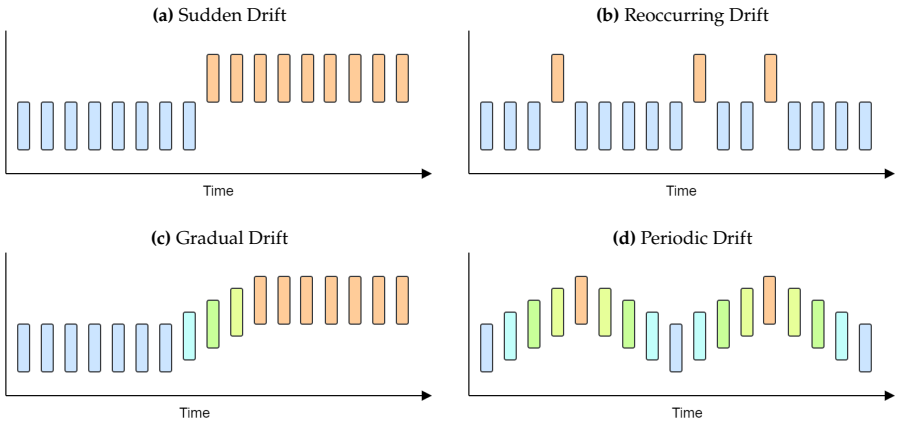


Fig. 4.1: A visualization of the characterization of the four primary types of concept drift.

- **Sudden Drift** occurs when the underlying data distribution suddenly and drastically changes. It leads to an abrupt shift in the relationship between input features and the target variable. Models that cannot adapt quickly might experience a significant drop in performance after such a drift event (Shown in Figure 4.1a). An example could be a sudden onset of heavy fog or a snowstorm, which can drastically reduce visibility as well as alter the appearance of people and objects in the scene.
- **Recurring Drift** involves the periodic occurrence of concept changes. The underlying data distribution may switch between multiple stable states over time, making it challenging for a model to settle on a single representation. This type of drift often requires continuous monitoring and adaptation to changing patterns (Shown in Figure 4.1b). The day/night cycle is an example of recurring drift, where there is a distinct recurring appearance change between day-time and night-time.
- **Gradual Drift** involves a slow and continuous change in the underlying data distribution over time. The target variable's statistical properties evolve gradually, leading to a gradual drift in the model's performance. Adaptation in this scenario requires methods that can slowly adjust to the

gradual shift in the underlying data distribution (Shown in Figure 4.1c). Surveillance systems observe evolving contexts such as construction sites would exhibit gradual drift as the surroundings gradually change and new structures are introduced and old ones are removed.

- **Periodic Drift** occurs when there is a recurring pattern in the drift events. It involves the repetition of concept changes at fixed intervals or time periods. Models that can detect and adapt to these regular patterns are better equipped to handle this type of concept drift (Shown in Figure 4.1d). Examples could include short-term periodic changes in the density of people due to sporting events or concerts.

Consequently, concept drift is a challenging aspect to handle as a combination of drift types can be present at any given time. Thus addressing concept drift encompasses two essential phases: Firstly, detection and assessment of the presence and magnitude of drift present. Secondly, adapting the system to address the observed drift.

Detecting Concept Drift

The first phase of addressing concept drift is to detect the occurrence of concept drift. The concept drift needs to be measurable which is challenging due to its contextual nature and because it is typically induced by unknown external factors [5, 43, 47]. Proxy's and approximations for quantifying concept drift often fall into one of three categories: error-rate-, distribution- and multiple hypothesis-based [43, 47, 77, 89, 90, 117].

Error-Rate approaches employ metrics similar or identical to those used to evaluate the system's primary task, to evaluate if a change in performance has occurred [30, 32, 76, 124]. Using the downstream task as the primary drift expectation metric lends itself to incorporating the performance tolerance expected during final deployment, however, it will require evaluation data to be annotated with the same granularity as the primary task. For fine-grained tasks, this can be particularly costly [43].

Distribution-based methods compare a given sample or window of samples of new data, with the known distribution. Concept drift is then present when a given sample falls outside of the known distribution [43, 50, 89]. Vision systems typically leverage encoder networks to condense a given sample into a dense vector representation for visual clustering [149, 152]. These methods additionally allow for the use of statistical- and distribution-based methods from other domains.

Multiple Hypothesis-based often employs a combination of external, Error-Rate- and/or Distribution-based methods [43, 89]. Employing multiple hypotheses instead of relying on a single drift detection model allows for drift detection along several parameters and across multiple windows. Ideally,

these approaches are leveraged when there is a need for very fine-grained detection. These are typically employed when there is a detailed understanding of the context and the types of drifts that can be expected [89].

Adapting to concept drift

When concept drift has been detected and has exceeded the maximum allowed tolerance, changes to the system must be made to adjust performance. Methods for adapting to visual concept-drift can be grouped into two methodologies: learning- and model-based Methods [43, 47].

Learning-based methods focus on integrating new knowledge through iterative updates to a given model, often manifesting as partial [38, 125, 133, 148] or complete [9, 38, 125, 133, 143, 148] retraining of the network. Which, depending on the chosen architecture, can quickly become a resource-intensive task. This is especially true for supervised methods, which would require annotated ground truths. In the adjacent field of domain-adaptation, self-supervised approaches [27, 98, 113, 119, 129, 137, 146] have become quite popular to refine an existing model on unlabeled data. Commonly, the effectiveness of these methods is evaluated based on the model's accuracy on other datasets, which share some commonalities with the chosen down-stream task [27, 98, 113, 146]. The effectiveness of these methods could serve as reliable methods to address the adapt to the challenges posed by concept drift. Model-based methods focus on training or inference of several specialized models, often in an ensemble setup [17, 90, 94, 142] where the contribution of a given model is controlled through handcrafted rules or fully learned [96, 125, 142, 148]. Ensemble approaches rely on being able to accurately detect the magnitude and type of drift to effectively adjust the model, or models, used for inference. Having a set of specialized models could provide increased performance by narrowing the required level of generalization, however often results in large levels of redundancy between models [142].

Thermal Object Detection in a Real-World context

Thermal-only object detection algorithms are woefully understudied, and models from the RGB domain are typically adapted directly to the thermal domain without architectural changes [2, 4, 62, 67, 76, 78, 104, 144, 153]. Historically, research has often applied YOLO variants [8, 36, 72, 76, 86, 140], Faster R-CNN variants [2, 65, 67, 104, 151] or SSD variants [2, 62, 153] to train thermal-specific models from scratch. Unlike in the RGB domain, large-scale thermal datasets are sparse, have very limited variation, or span very few days [6, 59, 109], which limits general evaluation of object-detection algorithms in the thermal domain. Thus it is very common that models train exclusively on data from the target domain [29, 75, 76, 104] making it difficult

to compare performance between proposed methods. Intuitively, the end-to-end learning scheme employed in training such algorithms should ensure the models learn domain-specific patterns [2, 76]. However, this assumption limits research on thermal-specific architectures, despite recent work indicating that thermal-specific architectures can be more computationally effective while displaying comparative performance [28]. Most research on thermal object detection involves leveraging multiple modalities [8, 17, 20, 136], and learning fusion methods that allow complimentary processing of said modalities. Commonly, this is achieved by having separate modality-specific feature-extraction networks and then fusing the latent representation [66, 73, 101, 154, 156, 157].

2.2 Thermal Object Detection in the Presence of Visual Concept Drift

The intersection between concept drift and thermal object detection presents interesting and unique challenges. This is particularly true in environments with high thermal variation. In these environments, it is difficult to learn a singular representation of the object, as not only does the visual appearance of objects change, but the environment also changes [6, 25, 59, 72, 76]. Existing object recognition datasets in the thermal domain span very short periods (typically short clips from a few select days) and have fairly uniform thermal conditions [6, 59, 109]. With the scope of these datasets it can be assumed that thermally, the sample typically falls within one of two distributions: daytime or night-time samples [72]. While this provides insight into performance across these assumed distributions, it fails to correctly address the gradual impact of concept drift that would be observed in a real-world context [89]. This is further exacerbated by datasets being captured by quantitative thermal cameras as opposed to qualitative thermal cameras, which are significantly more common in current real-world systems [41, 61].

Reducing the impact of concept drift on detection algorithms

Training on diverse thermal samples might serve to establish a good baseline for the performance of a given architecture in the thermal domain [76, 98, 136], but it does not facilitate modeling of the transitive states of concepts as they experience thermal concept drift. Some approaches attempt to capture representations between distribution by learning an ensemble of specialized models [17, 35, 90, 96]. Ensemble approaches tend to improve accuracy and reduce Miss-Rate (MR) but they also require significantly more resources to be run [141]. Using ensembles to reduce the impact of concept drift could be seen as a viable choice for deployment when a resource-efficient implementation and infrastructure surrounding a given model exist. Furthermore, deployments in resource-constrained environments could employ

model-selection methods to dynamically execute the ensemble at a lower resource cost [74, 99, 141]. Alternately some approaches address concept drift by incrementally improving the model, progressively integrating the drifting concepts in the training loop [98, 113, 126, 129]. While this serves quite efficiently at addressing the impact of incremental concept drift [126], it becomes quite problematic with reoccurring- or periodic- concept drift. With the continuous refinement of latent representations, it is prone to experiencing catastrophic forgetting, which is detrimental for long-term deployments [47].

The impact of long-term concept drift on the task of object detection is a niche topic, with thermal object detection further narrowing this field. The nature of thermal cameras lends itself as an ideal sensor type for outdoor environments. However, the current lack of research in addressing concept drift and the resulting performance degradation limits the possibility of widespread adoption.

3 Thermal Concept Drift during Long-term Deployment

In this section, the contributions and insights presented in Paper D are presented. Namely, this work proposed a novel real-world thermal dataset spanning 8 months (January to August), containing diverse weather conditions, human activities, and recurring cycles (weekdays, weekends, mornings, evenings, and seasonal changes). The presented dataset contains thermal video clips with associated meta-data (timestamp, temperature, humidity, precipitation, etc.) and serves as a platform for the evaluation of concept drift impact on various vision tasks. The dataset addresses a gap in the literature that could facilitate research into the understanding of concept drift-related challenges imposed during long-term deployments of computer vision algorithms, in the thermal domain.

3.1 Related Work

Studying the effects of concept drift in vision tasks often focus on a specific real-world use case or simulates drift by synthetically augmenting existing datasets [94, 100]. However, many large-scale datasets often span short periods [11, 48, 85, 88, 92, 111] or lack rich meta-data [1, 11, 25, 59, 85, 91, 92, 127], this is particularly true for thermal datasets [6, 59, 109]. Consequently, the evaluation of long-term thermal concept drift is not properly facilitated by existing datasets, leaving a critical gap in the literature. This also impacts the associated meta-data and the ability to identify contextual factors that could potentially be leveraged to identify concept drift factors. Urban environments play a pivotal role in the study of concept drift due to their high population

	Name	Year	Type	Duration	Period	Metadata
Stationary	UCSD [92]	2010	RGB	3.1	-	-
	Caltech Pedestrian [33]	2011	RGB	10	-	-
	VIRAT [105]	2011	RGB	29	-	-
	Avenue [88]	2013	RGB	0.5	-	-
	ShanghaiTech [85]	2018	RGB	3.6	-	-
	Surveillance Videos [127]	2018	RGB	128	-	-
	Street Scene [111]	2020	RGB	4	2 summers	-
	ADOC [110]	2020	RGB	24	1 day	-
	AU-AIR [11]	2020	RGB	2	-	Time, Positions
	MEVA [24]	2021	RGB/Thermal	144	3 weeks	GPS, Time
	LTD [102](Paper D)	2021	Thermal	298	8 months	GPS, Day/Night, Weather, Time
Changing	KAIST [59]	2015	RGB/Thermal	43.41	-	-
	CVC-14 [48]	2016	RGB/Thermal	11.8	-	-
	Oxford RobotCar [91]	2017	RGB/LiDAR	-	1 year	GPS, IMU, Day/Night, Weather
	Aachen Day-Night [118]	2018	RGB	-	-	GPS, Day/Night, Weather
	Gated2Depth [51]	2019	RGB/LiDAR	-	-	GPS, IMU, Day/Night, Weather
	Dark Zurich [114]	2019	RGB	-	-	GPS, Day/Night
	ACDC [115]	2020	RGB	-	several days	GPS, Weather
	Ford AV [1]	2020	RGB/LiDAR	-	1 year	GPS, IMU Day/Night, Weather, Time
	Bdd100k [150]	2020	RGB	-	-	Weather, Time

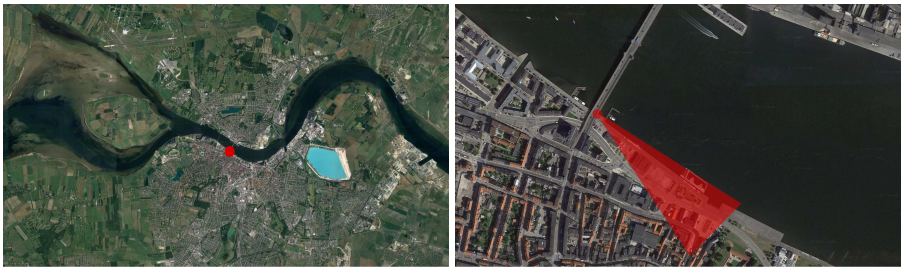
Table 4.1: *"Existing urban computer vision stationary and changing datasets. In the **Location** column 'changing' denotes a moving camera, like the ones on self-driving cars, whereas 'stationary' denotes static cameras, like the ones found in surveillance contexts. **Type** denotes the modality of the dataset (i.e., RGB, thermal, or LiDAR). **Duration** denotes the size of the dataset in hours. Whereas **Period** denotes the time span the data was captured in. Finally, **Metadata** denotes any additional information."* [102]. This table is adapted from [102], (Paper D)

density, primarily consisting of humans. Given that numerous systems are designed to enhance human security and well-being within urban settings, it becomes imperative that these systems operate with utmost efficiency.

"Notably, previous work in the field can be categorized into two primary types of datasets, wherein urban environments play a prominent role. The first type comprises datasets that feature scenes captured from stationary locations, such as those obtained from CCTV and surveillance cameras. These datasets are utilized for tasks like vehicle, pedestrian, and environmental detection and segmentation [24, 85, 111]. The second type consists of datasets where locations are constantly changing and are primarily designed for autonomous cars, robots, human egocentric footage, and anomaly detection [1, 59, 150].

As can be observed in Table 4.1, datasets used for autonomous driving typically feature changing locations and diverse modalities like LiDAR-, RGB-, depth-, GPS-, and IMU-data [1, 51, 118, 150]. They encompass longer-duration data, ranging from days to years, and focus on adverse weather conditions to enhance domain adaptation and robustness in autonomous driving and robotics applications. While thermal datasets are less common but still widely used, existing thermal datasets lack long-term data and rich meta-data which could be leveraged to identify impactful concept drift parameters.

Stationary datasets also lack duration information, and their relatively short duration limits their applicability in studying long-term effects on deployed



(a) Location of camera **(b)** Camera FoV

Fig. 4.2: Illustrations showing the location and viewing direction of the camera setup used to gather the data for the LTD dataset. satellite images captured using Google Maps. Location: Aalborg, Denmark. Map data ©2023 Google. Imagery: ©2023 Aerodata International Surveys, Airbus, Maxar Technologies, CNES / Airbus, Landsat / Copernicus.

machine learning solutions [33, 84, 85, 92]. With the absence of metadata, the study of concept drift is limited further. Most investigated datasets primarily concentrate on RGB data [11, 33, 85, 88, 91, 92, 105, 110, 111, 115, 118, 127, 150], with only a few containing both RGB and thermal data [24, 48, 59]. In compliance with the General Data Protection Regulation (GDPR), thermal imaging is preferable to preserve people's anonymity. This further eliminates the need for post-processing to protect personal data.

Furthermore, with the growing thermal imaging market [36], there is a pressing need for accessible long-term public thermal datasets.

3.2 The Long-term Thermal Drift (LTD) Dataset

Paper D introduced the LTD dataset. The dataset was collected using a stationary setup, observing the harbor front of Aalborg (as shown in Figure 4.2). The duration of the dataset spans two periods, with the first spanning May 2020 to September 2020, and the second spanning January 2021 to May 2021. In these periods a total of 298 hours were recorded on a Hikvision DS-2TD2235D-25/50 long-wavelength relative thermal camera [56]. Throughout the two periods clips of 2-minutes were recorded every 30 minutes, capturing activity at all hours of the day, as well as the gradual short-term changes (i.e., day/night), long-term changes (i.e., days, weeks, and months) as well as environmental changes (i.e., people, vehicles, rain, snow, fog, etc.). Examples of the visual variation can be seen in Figure 4.3.

With each clip the timestamp was stored, enabling pairing with external meta-data as well as identifying the time of day and season of the given clip. The Danish Meteorological Institute (DMI) provides an open-source weather API [60], which was used to extract weather-related meta-data. For each clip the following weather-related meta-data was obtained: Temperature

Fig. 4.3: *"Examples of extreme changes in the image data contained in the LTD dataset. The top half of the figure shows examples from February, March, April, June, and August respectively. These examples show the visual change which can be observed over time. The lower half shows examples of the drastic environmental changes that also appear in the dataset, in addition to the gradual change."* [102]. This figure is adapted from [102], (Paper D).

[$°C$], humidity [%], precipitation [kg/m^2], dew point [$°C$], temperature [$°C$], wind speed [m/s], sun radiance [W/m^2] and duration of sunshine [min]. An overview of average values can be seen in Table 4.2 and a detailed explanation of each meta-variable can be found on the API website [1].

Per month average for metadata

Month	Temp. [°C]	Hum. [%]	Precip. [kg/m^2]	Dew P. [°C]	Wind Dir. [$degrees$]	Wind Sp. [m/s]	Sun Rad. [W/m^2]	Sun [min]
Jan.	-0.48	90.10	0.01	-1.96	161.91	2.58	23.97	0.90
Feb.	-0.54	85.15	0.01	-2.83	131.00	2.95	51.12	1.42
Mar.	3.75	83.61	0.01	0.93	218.80	3.58	99.35	1.85
Apr.	4.47	97.25	0.13	4.10	126.50	2.97	67.31	2.23
May	10.74	75.46	0.01	6.07	217.32	3.04	256.76	3.66
June	16.36	71.46	0.01	10.57	151.27	2.37	256.46	3.63
July	12.91	75.32	0.01	8.46	268.15	3.97	270.17	3.62
Aug.	16.93	79.17	0.02	12.69	163.18	2.08	197.86	3.15

Table 4.2: *"Average metadata for each month. From left - temperature, humidity, precipitation, dew point, wind direction, wind speed, sun radiation, and minutes of sunshine in a 10-minute interval."* [102]. This table is adapted from [102], (Paper D).

[1] `https://confluence.govcloud.dk/display/FDAPI`

3.3 Investigating impact on vision tasks

The meta-data contains parameters that have previously been shown to have an impact on thermal cameras or directly affect the amount of IR radiation present. Existing research has shown that temperature and relative humidity impact various vision tasks such as concrete defect detection [134], temperature measurements [3, 63], and food inspection [49]. Precipitation and dewpoint temperature can be used to indicate the presence of, rain, fog, or condensation. These can directly increase the attenuation of the IR radiation [7, 23] and potentially indicate a build-up of moisture. Wet surfaces and buddles alter the reflectivity of the scene, which changes the amount of IR radiation reflected from heat sources [12]. Sun radiance and sunshine duration can cause rapid changes in the intensity of IR light, which in turn impacts the captured thermal images. Finally, wind speed and direction can introduce movement in the background elements of the scene, such as water ripples or ropes, as well as affect the camera's stability and position.

Experiments

Experiments were designed to investigate the impact of the aforementioned factors during long-term deployment as well as how they affect the amount of data required to achieve stable performance. Based on the temperature meta-data four months were selected for annotation. The coldest month (i.e., February) was selected as the training data, the test data would then include a similarly cold month (i.e., January), the median month (i.e., April), and finally the warmest month (i.e., August). Every clip was sampled at 1 fps, resulting in 120 frames per clip. The training set was further sub-sampled into three subsets containing data from the coldest day, the coldest week, and the entire month. For each of the subsets, 5000 and 100 frames were selected for anomaly detection and object detection respectively, using a greedy farthest point sampling method. Each frame was given a position in a 2D feature space based on its frame number and the associated temperature. These precautions were taken to ensure that each subset contained a fair and varied distribution of samples.

Anomaly detection was selected as the first task due to the use of autoencoders. Inherently the performance of autoencoders is tightly related to the training set, as they are trained to reconstruct a given input image. Concept drift would shift the data away from the training distribution thus limiting its ability to accurately reconstruct the image. A simple convolutional autoencoder, CAE, and two SotA autoencoders, VQVAE2 [93] and MNAD [106] were employed to provide a baseline. MNAD contains two variations, the first performs reconstruction of a given image, whereas the second takes a sequence of images and attempts to predict the subsequent frame.

Object detection was selected as the second task as it represents a fundamental vision problem that other tasks (such as tracking and re-identification) rely on. For this task, YOLOv5 and Faster R-CNN were chosen as benchmarking architectures. YOLOv5 was the latest iteration of YOLO at the time, and Faster R-CNN was a common baseline algorithm for object detection. Additionally, these two architectures have previously been successfully applied to outdoor thermal data [18, 45, 58, 76].

The resulting setup required each model to be trained on three separate subsets and all variations of each model would then be evaluated on the three test sets. The expected performance loss between the training data and the cold month would be expected to be low as their content is thermally similar, thus the cold month effectively serves as a measure of performance on unseen data and minimal if any thermal concept drift. However, the median and warm test sets would contain thermally distant samples and thus serve as two steps of increasing thermal concept drift.

3.4 Results and Insights

The impact of concept drift was quantified by the change in performance observed for the trained models. However, the evaluation is two-fold. Firstly the performance degradation is outlined. Secondly, the associated meta-data was used to identify weather parameters that displayed a significant correlation with the observed performance degradation.

Performance impact

As can be observed in Table 4.3, all tasks suffer a significant performance hit when testing on thermally distant samples. Regardless of the training data, all models exhibit a significant loss in performance when exposed to thermal drift. Notably, autoencoder performance increases significantly when a wider temporal span of data is provided for training.

This however is not observed for object detectors, who perform very similarly regardless of data on thermally similar months. which could indicate that that object detectors are fairly efficient at disentangling the objects from the background. Overall these results indicate that the performance of the models suffers significantly when exposed to thermal concept drift. However, it does not properly outline how the individual weather parameters contribute to the observed loss in performance.

Correlating Meta-Data and Concept Drift

When observing the examples presented in Figure 4.3, two primary causes for visual change can be identified. Namely, seasonal and day/night changes.

Model Performance Results

	Methods	Train Feb.	Test set Jan.	Test set Apr.	Test set Aug.
Anomaly Detection	CAE	Day 5k	0.0096	0.0202	0.0242
		Week 5k	0.0061	0.0167	0.0212
		Month 5k	0.0042	0.0109	0.0147
	VQVAE2	Day 5k	0.0051	0.0072	0.0068
		Week 5k	0.0039	0.0066	0.0061
		Month 5k	0.0021	0.0039	0.0035
	MNAD Recon.	Day 5k	0.0028	0.0057	0.0069
		Week 5k	0.0065	0.0066	0.0062
		Month 5k	0.0015	0.0041	0.0048
	MNAD Pred.	Day 5k	0.0008	0.0007	0.0009
		Week 5k	0.0007	0.0006	0.0007
		Month 5k	0.0007	0.0006	0.0007
Object Detect.	YOLOv5	Day 100	0.8010	0.5390	0.5240
		Week 100	0.7940	0.4540	0.4860
		Month 100	0.7930	0.4860	0.4830
	Faster R-CNN	Day 100	0.6760	0.3230	0.3370
		Week 100	0.6740	0.2790	0.3060
		Month 100	0.6400	0.2560	0.3180

Table 4.3: Results from performance study showing anomaly detection and object detection models. Results for anomaly detection are reported as average MSE, whereas object detection is reported as $mAP_0.5$. This table is adapted from [102], (Paper D).

Additionally, human activity can also play a significant role in visual changes. Thus, it can be inferred that the performance impact can be caused by either the weather conditions, human activity, or a combination of these two. Therefore, the density of people was included as a meta-parameter.

To identify whether parameters exhibit a correlation with performance, a basic Pearson's Correlation (PC) [128] and Distance Correlation (DC) [39] was calculated and the statistical significance was calculated with a p-value of 0.05. The F1 score was used to evaluate the object detection models as opposed to the more frequently used mAP [37, 82], as the F1 score provides a better measure of incorrectly classified cases. As can be observed in Table 4.4, the greatest correlation between meta-data and model performance is displayed by temperature and humidity, closely followed by time of day (i.e., day/night) and density of people in the scene. To discern whether the observed correlation is a sign of a causal relationship between the meta-data and model performance a Granger causality test was done to identify if there is a predictive causality between variables. Furthermore, to provide a more robust view of predictive causality two non-linear Neural Granger [130] tests were conducted in parallel with the normal Granger test [120].

		Performance / Meta-Data Correlation							
	Measure	Temp.	Hum.	Wind Dir.	Wind Sp.	Precip.	Activ.	D./N.	Hour
CAE	P. C.	0.679	0.636	0.018	0.157	0.109	0.270	0.545	0.166
	D. C.	0.682	0.588	0.158	0.170	0.126	0.291	0.538	0.287
VQVAE2	P. C.	0.381	0.690	0.001	0.194	0.172	0.217	0.403	0.124
	D. C.	0.347	0.639	0.174	0.201	0.224	0.217	0.382	0.213
MNAD Recon.	P. C.	0.607	0.672	0.016	0.173	0.126	0.220	0.509	0.156
	D. C.	0.617	0.629	0.188	0.177	0.155	0.252	0.501	0.273
MNAD Pred.	P. C.	0.107	0.277	0.064	0.152	0.072	0.677	0.369	0.137
	D. C.	0.231	0.348	0.154	0.172	0.086✗	0.665	0.462	0.312
YOLOv5	P. C.	0.261	0.258	0.102	0.011	0.096	0.124	0.047	0.009
	D. C.	0.293	0.283	0.146	0.094	0.135	0.255	0.113	0.174
Faster R-CNN	P. C.	0.354	0.456	0.115	0.135	0.0124✗	0.199	0.147	0.001
	D. C.	0.334	0.460	0.228	0.149	0.065	0.231	0.163	0.118

Table 4.4: *"Correlation between the model's measured performance values MSE and F1-score and the weather, time, and scene activity features. Two correlation measures are used - Pearson's (P.C.) and Distance (D.C.) correlation. Measures that do not meet the statistical significance threshold of their p-values are shown in red. The Day/Night features are specified as D./N."* [102]. Autoencoders (i.e CAE, VQVEA2, and MNAD variants) have their correlation calculated based on MSE, whereas object detectors (i.e., Faster R-CNN and YOLOv5) have their correlation calculated based on F1-Score. This table is adapted from [102], (Paper D).

Interestingly the results of the Granger tests (shown in Table 4.5), show that people density in the scenes displays no predictive causality with performance, despite showing a correlation. This would hint at the correlation being the result of a second-hand relationship. Intuitively, this makes sense as people typically move outside in higher densities in the daytime, particularly in 'good' weather. Additionally, it can be seen that changes in temperature and humidity display high predictive causality for the autoencoder models. In contrast, the object detection models seem to display a correlation between similar meta-data variables and performance as autoencoders, however, they do not share the same predictive causality. Finally, the time of day (i.e., day/night) indicator shows strong predictive causality with model performance across the board (with the exception of Faster R-CNN).

3.5 Summary and Contributions

During the long-term deployment of thermal vision systems, it is almost certain that visual concept drift will be observed. Changes in environmental conditions can be correlated with a degradation in the performance of the deployed system, making it a particularly vital challenge to overcome. While only a few weather parameters (i.e., temperature, humidity, and day/night) have predictive causality with the performance of anomaly detection and object detection on this particular dataset. It highlights the need for more research

	Predictive Causaility of Meta-Data											
	Temp.			Hum.			Activ.			D./N.		
	Basic	LSTM	MLP	Basic	LSTM	MLP	Basic	LSTM	MLP	Basic	LSTM	MLP
CAE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
VQVAE2	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
MNAD Recon.	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
MNAD Pred.	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
YOLOv5	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗
Faster R-CNN	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓

Table 4.5: *Results from calculating linear and non-linear (LSTM and MLP) Granger causality tests. The cells marked with ✓ show positive predictive causality, while cells marked with ✗ show no significant causality.* [102], Autoencoders (i.e CAE, VQVEA2, and MNAD variants) have their predictive causality calculated based on MSE, whereas Object Detectors (i.e., Faster R-CNN and YOLOv5) have their predictive causality calculated based on F1-Score. This table is adapted from [102], (Paper D).

and development of methods resistant to the impact of concept drift. The introduction of the LTD dataset and the experiments conducted herein serves to establish a benchmark and facilitate the study of thermal concept drift in future research.

In this section, the work conducted in Paper D was presented. In large parts, the work focused on addressing the impact of concept drift in the thermal domain, identifying weather-related components which directly influence the performance of thermal vision systems during long-term deployment. The contributions of the work described herein can be summarized as follows:

- The introduction of a novel large-scale thermal dataset spanning eight months containing over 298 hours of video, with rich meta-data providing information about the time, weather, and thermal conditions present in each clip. This is currently the largest publicly available thermal dataset for concept drift analysis.
- Established baselines for evaluation of thermal concept drift with diverse meta-data, allowing for highly granular studies and analysis of the long-term deployment of thermal vision systems.
- Experiments investigating the impact of weather-related factors during long-term deployment, by analyzing the performance impact of anomaly detection and object detection models. Highlighting that models which produce fine-grained predictions are especially sensitive to thermal concept drift.
- In-depth analysis of correlation and predictive causality between available meta-data and model performance. Identifying key conditions

which impact the performance of vision tasks and contribute to thermal concept drift.

4 Evaluating Long-term Robustness under Concept Drift

In this section, the contributions and insights presented in Paper E are presented. Namely, the extension of the LTD dataset, the results of the "Seasons in Drift" challenge, and the extended insights into the impact of weather factors on object-detection algorithms.

At the 2022 ECCV Workshop "Real-World Surveillance: Applications and Challenges (RWS@ECCV)" an extension to the LTD dataset was introduced paired with a fine-grained object-detection challenge[2]. The goal of the challenge was to invite the research community to test their approaches with known concept drift, and present novel approaches to combat the impact of concept drift. In this section we will outline the contents of the dataset extension, the experiment setup and evaluation method, results, and trends.

4.1 Related Work

The impact of long-term concept drift has been a prominent topic for assessing the potential for real-world performance of computer vision algorithms [54, 64, 147, 159]. However, in the field of object detection, there is a distinct lack of datasets spanning long-term deployments. On this topic, the closest related work (excluding Paper D [102]) which span long periods of time and contain known concept drift are ACDC [115] and Ford AV [1] which both are dynamic autonomous driving datasets. Furthermore, while they span great periods the duration of data and sample rate of those durations are short and infrequent, making it difficult to assess the impact of gradual concept drift. With the introduction of the LTD dataset [102] (Paper D) a benchmark had been established for evaluating and analyzing the impact of thermal concept drift on object detection algorithms. The prior experiments only featured two object detectors (i.e., YOLOv5 and Faster R-CNN), and 4 months' worth of data with limited object annotations. While the experiments showed the drastic impact thermal drift could have on these tasks, there was a lack of granularity when trying to evaluate long-term performance. Rather than proposing a novel dataset, extending an existing one, in particular one with long-term consistent data will serve to further provide avenues of comparisons and study. Notably, expanding the LTD dataset with rich annotations would serve to facilitate more

[2] https://chalearnlap.cvc.uab.cat/challenge/51/description/

detailed studies on thermal concept drift, and provide existing work avenues for more detailed analysis.

4.2 Extended Object Annotations

To allow for granular analysis of thermal object detectors under thermal drift on the LTD dataset, it was clear that the dataset needed to be extended significantly. Instead of a specific number of annotated for each subset, the entire dataset was sampled at 1 Frames Per Second (FPS), resulting in 1.069.247 frames in need of annotation. A total of 6.868.067 objects were annotated, extending the LTD dataset object annotations by an order of magnitudes (an overview of annotations can be seen in Table 4.6). As the images were annotated with no discrepancy the extended dataset naturally display similar object activity, density, and conditions as would be expected in a real-world scenario.

	New Annotations	
Frames	Total	1.069.247
	w/ objects	224.609
	w/o objects	844.638
Objects	Total	6.868.067
	Unique Objects	143.294
	Person	584.139
	Bicycle	293.280
	Motorcycle	32.393
	Vehicle	701.255

Table 4.6: Overview of annotations added to the LTD dataset. This table is adapted from [102], (Paper D)

Notably, this means that the vast majority of frames contain no objects of interest. While the empty frames were not leveraged for evaluation they were provided they could potentially be leveraged for some approaches that leverage scene-priors [70]. The surveillance context and point-of-view of the camera also ensured that the size of objects was quite small due to their distance from the camera. Following the MS COCO classification of object sizes, i.e., $area < 32^2$, $32^2 < area < 96^2$, $area > 96^2$ for small, medium and large objects respectively, the vast majority of all but vehicle objects would be classified as small objects (a detailed overview can be seen in Table 4.7 and visualized in Figure 4.4). The distribution predominantly consisting of small objects further adds to the challenge of the dataset, as small objects are notoriously difficult for object detectors [19, 71, 103, 160].

4.3 Experiment Setup

Following the protocol established in Section 3.3, the challenge was separated into three different tracks, i.e., the *Day*-, *Week*-, and *Month*-track. Each track employed the same evaluation protocol, test data, and performance metrics, however, they differed in the amount of training data. For evaluation, the data from January, March, April, May, June, July, August, and September, was split into equal parts validation- and test subsets.

	Class-wise Object Frequency (Area)			
	Person	Bicycle	Motorcycle	Vehicle
Small	5.663.804	288.081	27.153	113.552
Medium	454	7	0	37.007
Large	176.881	5.192	5.240	550.696
Total	5.841.139	293.280	32.393	701.255

Table 4.7: This table summarizes observed object frequency with respect to the total area of objects. Categorization follows the MS COCO size classification scheme, i.e., $area < 32^2$, $32^2 < area < 96^2$, $area > 96^2$. This table adapted from [68] (Paper E)

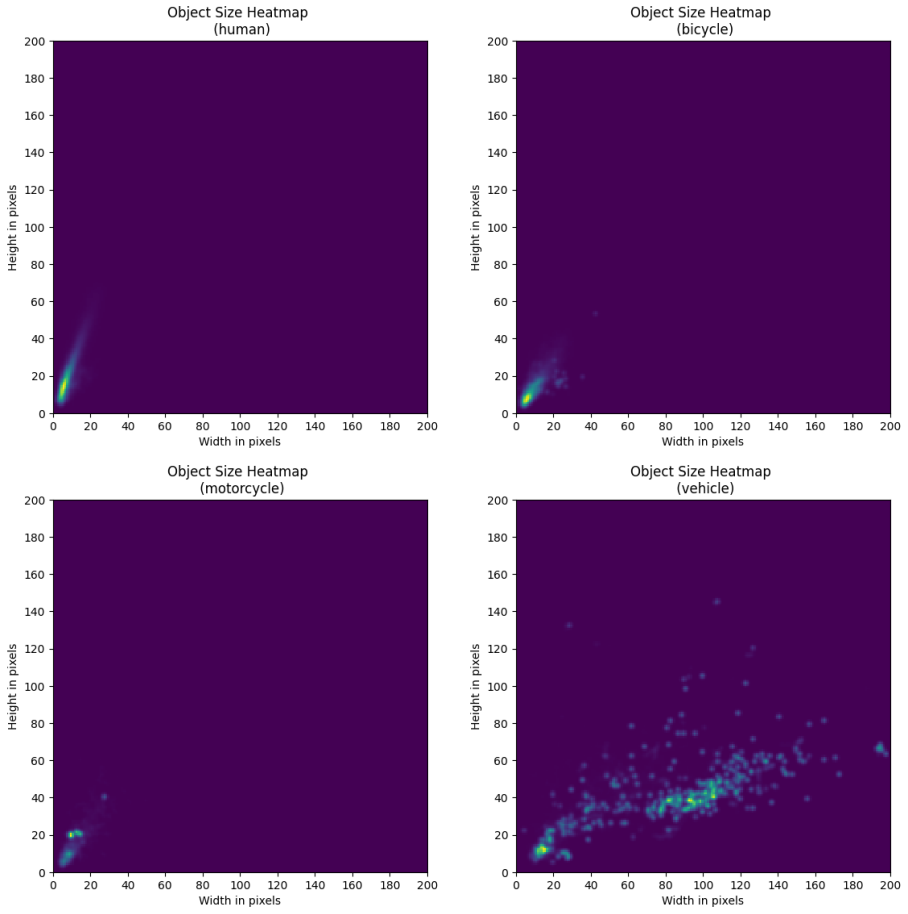


Fig. 4.4: Visualization of class-wise object-size frequency. The illustrations in this figure break down the general distribution of objects in terms of their height (Y-axis) and width (X-Axis). This figure is adapted from [69], Paper F

The experiment would then be executed in two phases: the development phase, and the test phase. During the development phase, the training data and annotations were made available to the participants, who were required to submit their predictions based on a validation set. Subsequently, during the test phase, participants were tasked with submitting their results for the test data. The test data was made available only a few days before the challenge's conclusion to prevent manual annotation or similar attempts at over-fitting the test set. At the conclusion of the challenge, participant rankings were determined using the test data. Primarily the competition centered on the submission of results, but to be eligible to win the participants were obliged to share their codes and trained models after the challenge's conclusion to facilitate result replication. After the deadline and submission of the training code as well as pre-trained models, the submissions were manually verified. Following the challenge, submissions were considered valid if their top-ranked methodologies successfully passed the code verification stage.

Due to the significant increase in annotations, the existing object detectors would not compete on equal footing with the newly extended data, thus the best-performing model (i.e., YOLOv5) was retrained from scratch on the new data. In addition to serving as a baseline performance, it was used to produce an approximation of difficulty for each month. The difficulty of each month was based on the degradation in performance exhibited by the baseline model.

Evaluation Method

The performance of the baseline model on the January subset ($base_{Jan}$) was given the initial weight of 1.0. The weight of subsequent subsets was calculated as $w = base_i/base_{Jan}$ where i refers to a given subset. The resulting weights ranged from $1.0 - 1.93$. To avoid accuracy exceeding 100% the weighting was further remapped to $0.75 - 1.0$. When submitting results the evaluation script separated samples into their respective subsets and calculated a subset specific mAP. The final performance metric was then calculated as a weighted average of the subset-specific performance and their respective weighting. The performance for each month was measured in mAP across all classes at $.50 : .05 : .95$ Intersection over Union (IoU), following the evaluation metric used for other large-scale object-detection datasets [82] and extending the evaluation criteria of [102] (Paper D).

Intuitively, models who performed better on more difficult months (i.e., samples that experienced more drift) would be worth more in the combined score. The aim was to incentives methods that generalized well or employed some mechanism to combat concept drift.

4.4 Results and Insights

To obtain performance insights for each model and identify potential advances or methods that display great promise at combating thermal concept drift analysis was conducted as a two-step process. The initial step is the performance measurement used to evaluate participant performance (as seen in Table 4.8), i.e., per subset performance as well as a weighted performance score. Secondly, the performance of the model that passed the code verification stage was computed with respect to temperature, humidity, object size, and object density (shown in Figures 4.5a to 4.5c).

Leaderboard

As can be observed in Table 4.8, top participants across all tracks managed to obtain significant increases in performance across all months when compared to the baseline model. Notably the winning submission also used a YOLOv5 variant, whereas the runner-up employed a cascaded transformer model. Unfortunately, very few non-winning participants refused to submit

Participant	mAP_w	mAP	Jan	Mar	Apr	May	Jun	Jul	Aug	Sep
Track 1 (day level)										
Team GroundTruth*	**.2798**	**.2832**	.3048	**.3021**	**.3073**	**.2674**	**.2748**	**.2306**	**.2829**	**.2955**
Team heboyong*	.2400	.2434	**.3063**	.2952	.2905	.2295	.2318	.1901	.2615	.1419
Team BDD	.2386	.2417	.2611	.2775	.2744	.2383	.2371	.1961	.2365	.2122
Team Charles	.2382	.2404	.2676	.2848	.2794	.2388	.2416	.2035	.2446	.1630
Team Relax	.2279	.2311	.2510	.2642	.2556	.2138	.2336	.1856	.2214	.2235
Baseline*	.0870	.0911	.1552	.1432	.1150	.0669	.0563	.0641	.0835	.0442
Track 2 (week level)										
Team GroundTruth*	**.3236**	**.3305**	.3708	.3502	**.3323**	.2774	**.2924**	**.2506**	**.3162**	.4542
Team heboyong*	.3226	.3301	.3691	.3548	.3279	**.2827**	.2856	.2435	.3112	.4662
Team Hby	.3218	.3296	.3722	.3556	.3256	.2806	.2818	.2432	.3067	**.4714**
Team PZH	.3087	.3156	**.3999**	**.3588**	.3212	.2596	.2744	.2502	.3013	.3592
Team BDD	.3007	.3072	.3557	.3367	.3141	.2562	.2735	.2338	.2936	.3942
Baseline*	.1585	.1669	.2960	.2554	.2014	.1228	.0982	.1043	.1454	.1118
Track 3 (month level)										
Team GroundTruth*	**.3376**	**.3464**	**.4142**	**.3729**	**.3414**	**.3032**	**.2933**	**.2567**	.3112	**.4779**
Team heboyong*	.3241	.3316	.3671	.3538	.3289	.2838	.2864	.2458	**.3132**	.4735
Team BDD	.3121	.3186	.3681	.3445	.3248	.2680	.2843	.2450	.3062	.4076
Team PZH	.3087	.3156	.3999	.3588	.3212	.2596	.2744	.2502	.3013	.3592
Team BingDwenDwen	.2986	.3054	.3565	.3477	.3241	.2702	.2707	.2337	.2808	.3598
Baseline*	.1964	.2033	.3068	.2849	.2044	.1559	.1535	.1441	.1944	.1827

Table 4.8: *"In this table the leaderboard of the ECCV - ChaLearn Seasons in Drift Challenge are shown. Top solutions are highlighted in bold, and participants that passed the code-verification stage are marked with a '*'."* [68]. This table is adapted from [68] (Paper E)

code for verification thus preventing a broad in-depth benchmark and analysis of the various models. Contrary to initial benchmarks on the LTD dataset, the difference in training data provides a significant increase from *day-* to *week-*, and further a slight increase with the *month*-subset. While this trend is consistent across all subsets, the change relative to the original benchmark implies that intelligent sampling a diverse training set, is especially important when the annotation budget is restricted.

Interestingly, the performance on subsets at either end of the extremes also displayed the best performance, where it would be expected to gradually degrade. This could indicate that during long-term deployments the contents of the thermal scene could be grouped into two distinct distributions. This is further supported in Table 4.4, where day/night is shown to be a highly correlated aspect. Whether this is in optimal solution or just a product of generalization or model-capacity of the object detectors would need to be further investigated.

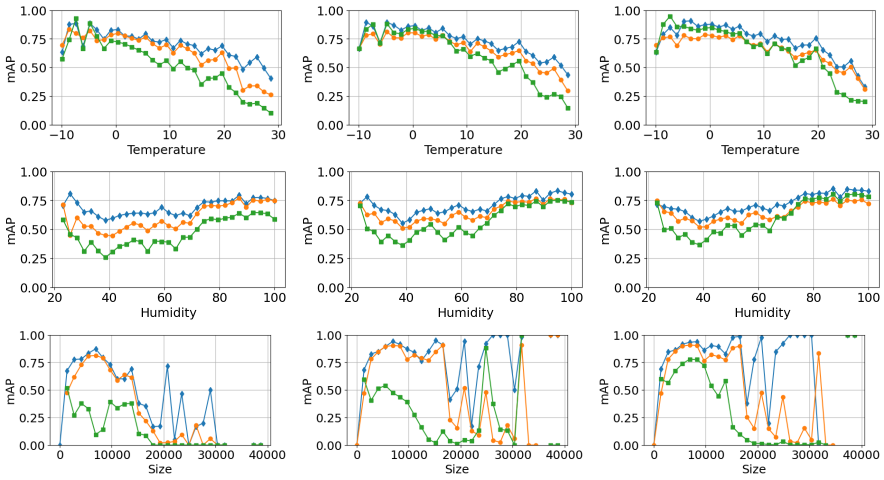
Weather Analysis

The intuition that there is a correlation between changes in temperature and performance (as detailed in Section 3.4) is further supported by the degradation observed in the Figures 4.5a to 4.5c. Where it can be observed that the top two models experienced consistent degradation with respect to temperature across all three tracks.

Interestingly, humidity partially follows a similar trend, when looking at Table 4.2 it can be observed that the average humidity drops when progressing from the colder to the warmer months. However, the observed performance impact is much more inconsistent, particularly with very low humidity not presenting a significant performance impact. This could further be a sign of models generalizing to a set of representations, favoring extreme conditions rather than modeling a smooth transition.

The impact of object sizes became very unpredictable at larger object sizes, most likely as a result of the low sample size both during training and evaluation. However, it can be observed that the baseline model in particular heavily degrades as it is exposed to larger objects. Intuitively, the model's reliance on anchor points results in the models fitting their reference anchor points to the most common object size and shape.

Similar to humidity performance of models remains fairly consistent at intermediate densities, and drops significantly at the extremes of each distribution. Notably, there is a consistent increase in performance slightly prior to the extremes. It is unclear what caused this, however, intuition would indicate that it is a product of "easy" samples in a sparse segment of the distribution.



(a) Day Track **(b)** Week Track **(c)** Month Track

Fig. 4.5: Visualization of model performance with respect to temperature, humidity, and object sizes, for day (Figure 4.5a), week (Figure 4.5b), and month (Figure 4.5c) tracks. This figure was adapted from [68], (Paper E).

Notable Participant Architectures

The winning participant and runner-up both employed significant methods for achieving their respective performances. Particularly the winning participant is worth highlighting as they employed methods surrounding the baseline model [53, 139, 142] to improve its performance in general as well as robustness to concept drift. During the training they sparsely sampled the training set into several subsets, these subsets were employed to train several models which each overfitted to their respective data distribution. With these specialized models, they employed a "model soup" [142] to merge the weights into a singular model. The key behind model soups is that the specialized models are able to learn filters that are otherwise suppressed during generalization, and by combining the weights of the specialized models a more robust general model can be obtained [142]. They further employed the unlabeled video frames prior to predicting sequences of bounding boxes, this sequence of predictions was then refined by a learned sequential Non-Maximum Suppression module [53]. This allowed the model to leverage temporal data to relate objects temporally and predict classes of detected objects based on temporal consistency.

The runner-up in large parts addressed the problem with compute and expansive data augmentation [10, 13, 44, 122, 155]. Notably they employed a SotA SWIN-Transformer [87] in a cascaded backbone configuration [80], with cascaded detection-heads [14]. several backbone networks were leveraged to progressively refine the input of the network and produce semantically

meaningful features [80]. However, due to the sheer scale of the resulting backbone, they potentially contain a lot of redundant information. To combat this the auxiliary detection heads were used to assist the backbone in learning meaningful representations at every level.

4.5 Summary and Contributions

The impact of concept drift on thermal object detection still remains a significant challenge for the viability of automated vision systems, the extension of the LTD dataset can help facilitate further research and development of object-detection algorithms that are able to adapt to the thermal concept drift observed during long-term deployment. To encourage research into this topic a public challenge was issued for the "Real-World Surveillance: Applications and Challenges" Workshop at ECCV2022. Participants took various approaches to address the challenge of concept drift and succeeded in improving performance over baseline. Notably, it is possible to outperform large compute-heavy SotA transformers by employing intelligent design surrounding the training and prediction of lightweight models.

In this section, the work conducted in Paper E was discussed. In large parts, the work focused on an extension of the object annotations of the LTD dataset, inviting the research community to develop object-detection algorithms robust to concept drift and analyzing trends. The contributions of the work described herein can be summarized as follows:

- An extension to the novel LTD data, expanding the object notation by an order of several magnitudes, making it not only the largest single-scene dataset for the study of concept-drift but also the largest single-scene thermal person detection dataset with over 6.8 million annotated objects.
- Conducted a public challenge, inviting the research community to participate in developing object-detection algorithms with the aim to combat the impact of thermal concept drift on object detectors. Establishing baselines for complex SotA thermal object-detectors under long-term concept drift.
- Analyzed the impact of model performance with respect to various weather, seasonal and object-centric parameters, providing insights into methods aimed at robust object detection under thermal concept drift.

5 Training Weather-aware Detection Algorithms

In this section, the contributions and insights presented in Paper F are Discussed. The dynamic challenge of concept drift further finds a compelling connection with contextual awareness. As the world evolves and changes over time necessitating adaptation of concepts and pattern recognition, contextual clues could help guide algorithms towards informed adaptation to the observed drift [16, 72]. Contextual awareness in that sense could empower computer vision systems to become more resilient to concept drift. Particularly by understanding the context the system would be able to discern if the visual change is a result of concept drift or contextual variation [16, 123]. Particularly for long-term deployment in outdoor environments, adverse weather is an especially challenging topic as it potentially obscures vision in conjunction with inducing sudden concept drift [47].

5.1 Related Work

Contextual awareness has been shown to improve the performance of object detection systems [16, 57, 72, 86, 86, 95, 97, 107, 108, 123, 138], and while research has been sparse in the thermal domain, the general improvement they pose could potentially be transversal between RGB and the thermal domain. As shown in [102] (ThesisPaper D), changes in weather conditions can in some cases be strongly correlated with concept drift for thermal videos. While the literature on context-aware object detection has primarily focused on context-awareness in the spatial sense [83, 86, 97, 108, 123, 138], weather-aware approaches have seen increasing interest in recent years [8, 16, 57, 72, 86, 95, 107]. Methods that leverage spatial context-awareness have shown great promise by leveraging global information to make informed local decisions, and accurately improve detection of heavily occluded objects [97, 123, 138]. Similarly, resilience towards adverse weather conditions can be achieved by leveraging global information to inform an understanding of image degradation induced by the weather [16, 57, 72]. Weather-aware methods can be roughly grouped into two groups of approaches: firstly, approaches that leverage awareness of contextual information (e.g. weather conditions) to learn robust representations that are agnostic to adverse weather conditions [57, 72] and thus do not impact the performance of the chosen network at inference time. Secondly, approaches that directly leverage mechanisms to adjust the weighting of internal computations based on weather features [8, 16].

Some approaches aim to refine the representations learned by the network to be robust to different perturbations [57, 72, 76]. Typically, done by employing an additional optimization goal [57] or auxiliary task [72] which can be removed during inference. By including weather awareness in the training loop, the algorithm can effectively be guided toward stronger representations.

In cases where exact knowledge is vital, the model can be made aware by predicting the weather in addition to its primary task. Effectively infusing intermediate representation with weather information [57, 72]. Employing these methods previous work has shown significant performance increases in adverse weather conditions [57, 72, 76].

Lastly, while not addressing concept drift or weather awareness directly, the "translation" approaches proposed in [86] and [140] could also potentially alleviate the impact of concept drift while being contextually aware of the contents. They do this by essentially including a "translation" layer that encodes and decodes the input image into a representation where the adverse weather is removed. Essentially, by synthetically adding simulated adverse weather to your training sample, you can teach a smaller "translation" network to reconstruct the image without the synthetic degradation [140].

5.2 Weather Aware Conditioning

Weather-aware conditioning of object detectors is a novel approach to improving the prediction accuracy of object detection systems, through leveraging contextual information to alter the behavior of the network. Traditional approaches often leverage a binary classification of contextual clues as a method to induce weather-aware knowledge into the model [57, 72]. This might be ideal for synthetic datasets and real datasets captured with absolute-thermal cameras

This approach relies on assumptions about the underlying data distribution or distributions, that may not hold in real-world scenarios. In an uncontrolled environment, there are likely various unknown variables that could introduce noise to the signal, making it challenging to accurately distinguish ground truth close to the bin edges [52, 81]. Posing the problem as a continuous prediction problem rather than a discrete one could allow the network to model representations that better fit the gradual nature of concept drift. However, as discussed previously in Section 3.1 and shown in Section 3.2, very few stationary datasets have meta-data with the level of granularity that would allow continuous prediction. The LTD dataset favorably contains such a level of resolution with its metadata, making it an ideal dataset for studying context-aware conditioning of thermal-object detection algorithms. As the thermal images in the LTD dataset are recorded with a relative thermal camera, even scenes with similar meta-conditions may exhibit slight visual differences. Consequently, predicting exact values from visual data becomes an ill-posed problem due to the inherent noise induced in the signal. To address this issue a predefined degree of deviation is baked into the optimization of the weather prediction component.

Conditioning Methods

Two distinct approaches for incorporating weather-conditioning into object detection models were explored. The first method, referred to as 'Direct Conditioning', involves directly integrating weather information into the latent representation of predictive branches through a conditioning layer proposed by [72]. Originally, the weather-conditioning element, derived from an auxiliary classification network, aimed to discern day-time and night-time distributions, making the network "aware" of weather variations. In contrast, the second approach referred to as 'Indirect Conditioning', leverages vision-transformers and their self-attention mechanism to encode information from the input image into a weather token. By utilizing its transformers' global reasoning this method allowed the network to dynamically disregard weather-related relationships that do not contribute meaningfully.

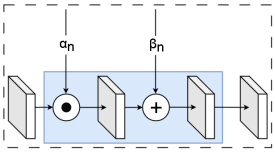


Fig. 4.6: Visualization of the conditioning layer proposed in [72]. This figure was adapted from [69] (Paper F)

YOLOv5: Direct Conditioning

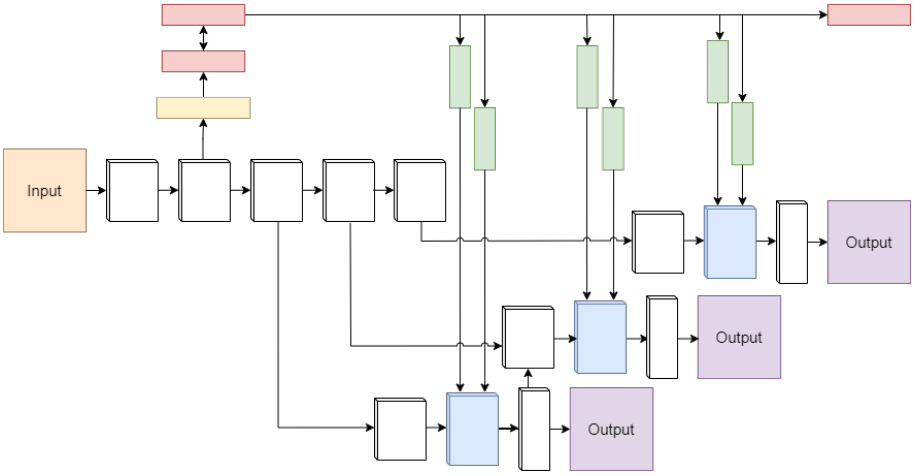


Fig. 4.7: *"YOLO-styled direct-conditioning network. Red, blue, green, and yellow denote the auxiliary branch-, conditioning layer-, conditioning layer, feed-forward network, and pooling layers respectively."* [69]. Blue boxes represent conditioning layers, This figure was adapted from [69] (Paper F)

The direct conditioning variant expanded on the initial configuration proposed in [72]. By leveraging the intermediate representation of the auxiliary branch the detector's intermediate representation is directly conditioned

thereby incorporating weather information into its semantically rich representation. The task of the auxiliary branch is to predict the current weather condition and thus would inform the detection branches' predictions. The YOLOv5 model [135] served as the original implementation. An auxiliary branch was extended from one of the early stages of the feature extractor, with fully connected layers condensing the representation. This condensed representation was then fed to a prediction head, generating a single value regressed using a novel L1 loss. Individual fully connected layers conveyed the representation to the conditioning layer in different stages of the network, before each stage's prediction head. The conditioning layer, shown in Figure 4.6, consisted of an element-wise multiplication and summation with the auxiliary representations.

Deformable DETR: Indirect Conditioning

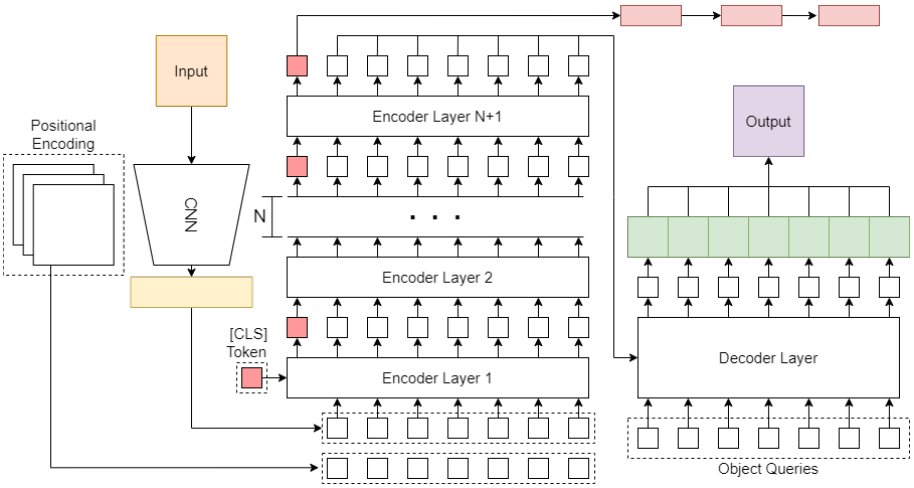


Fig. 4.8: *"DETR-style transformer network with indirect conditioning. Red, light-red, blue, green, and yellow denote the weather-token, auxiliary branch-, feed-forward network, and embedding layers respectively."* [69]. This figure was adapted from [69], (Paper F).

The in-direct conditioning variant extends deformable-DETR [158] with a learnable classification token, utilizing its encoding to predict the auxiliary task of weather condition prediction. Despite the possibility of the transformer learning embeddings optimized for affinity with the classification token, the network is designed to disregard weather-related embeddings when they do not significantly benefit optimization. Unlike the directly-imposed approach, this network can dynamically ignore image regions that do not provide relevant contextual information. Drawing inspiration from the use of a [CLS] token in the original BERT paper [31] and aggregation trends presented in [121], an

additional learned token was added to the input to progressively aggregate global weather-related information. This enables continuous aggregation of global information from the input to a single representation (i.e., token). Before the decoder, the [CLS] token was separated and directed to an auxiliary branch, consisting of fully connected layers, to map the weather token to a single value for regression.

Training Setup

The auxiliary prediction head in both the direct and indirect variants was trained in a supervised manner to guide them to accurately predict observed weather phenomena. This meant that the original data split used in Paper D [102] and Paper E [68] contained insufficient variation in the targeted weather conditions. As such the data was split into three equally sized train-, validation- and test- subsets, spanning the entire dataset. With the training set now containing a large amount of variation both in terms of meta-data and visual appearance the previous benchmarks would not reflect the performance that could be expected by the baseline models given the more diverse subsets. Thus each variant had a baseline model trained from scratch following the methodology described in their respective implementation/paper [135, 158].

Due to the inconsistent ranges of the meta-data (i.e., $[-13, \ldots, 28]$, $[0, \ldots, 100]$, $[0, \ldots, 24]$= for temperature, humidity, and time of day respectively), the impact of the absolute distance based loss would also be inconsistent and thus could cause unstable variances [15, 34, 112]. To address this the ranges were all remapped to $[-2, \ldots, 2]$ during internal computations, and then subsequently remapped back to their original value for later analysis.

For each type of conditioning scheme, several models were trained and conditioned on a set of meta-data. Each variant thus resulted in 4 models: a baseline and three

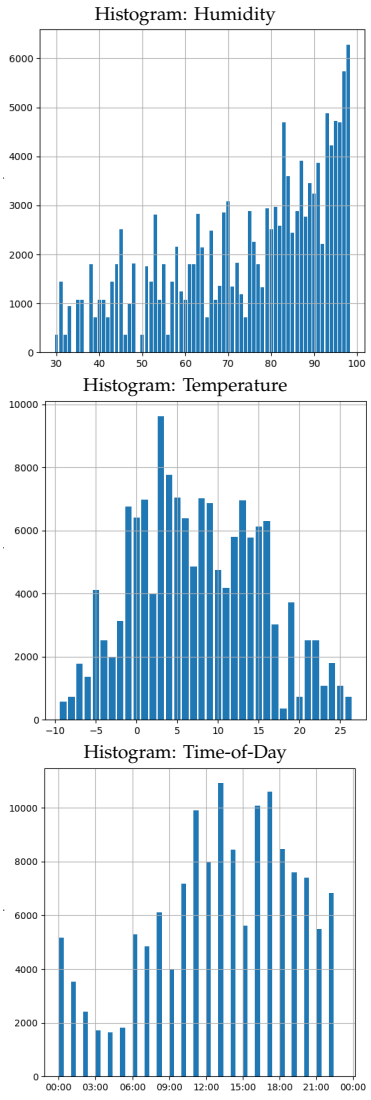


Fig. 4.9: Histograms of meta-data variables. This figure is adapted from [69], Paper F

models conditioned on temperature, humidity, and time of day respectively. During training, the losses were exponentially reduced within, $\pm 5°C$, 10%, and $1hour$ deviation from the ground truth. Thus the accuracy of the prediction head could be expected to display a similar level of mean Average Error (mAE), whereas the Standard Deviation (St.Dev.) can be used to assess the consistency of the deviation.

5.3 Results and Insights

Model	mAP_{voc}	mAP_{coco}	mAP_L	mAP_M	mAP_S	MR
YOLOv5 (Baseline)	**0.604**	**0.465**	**0.825**	**0.640**	0.491	0.342
YOLOv5 (Pretrain)	0.600	0.454	0.831	0.621	0.489	0.324
YOLOv5 (Temp.)	0.584	0.410	0.796	0.590	0.468	0.322
YOLOv5 (Hum.)	0.493	0.293	0.675	0.560	0.268	0.357
YOLOv5 (ToD)	0.549	0.439	0.805	0.566	0.431	0.356
DN-DETR Baseline	0.378	0.348	0.123	0.344	0.563	0.421
DN-DETR (Temp.)	0.225	0.148	0.100	0.190	**0.682**	0.389
DN-DETR (Hum.)	0.191	0.132	0.100	0.160	0.671	0.415
DN-DETR (ToD)	0.219	0.142	0.00	0.169	0.661	0.410
Def. DETR Baseline	0.332	0.202	0.005	0.051	0.637	0.383
Def. DETR (Temp.)	0.297	0.184	0.001	0.045	0.620	**0.351**
Def. DETR (Hum.)	0.213	0.114	0.000	0.020	0.517	0.416
Def. DETR (ToD)	0.289	0.178	0.001	0.040	0.619	0.395

Table 4.9: "*In this table the mean Average Precission (mAP), and Miss-Rate (MR) of direct- (YOLOv5) and indirect-conditioning (DETR) variants are detailed. Highlighted with* ***bold*** *is the best performing across all models and highlighted with underline is the best performing model for a given architecture.* mAP_{VOC} *denotes mAP where IoU is atleast 0.5,* mAP_{COCO} *denotes mAP at varying IoUs (i.e., {0.50, 0.55, 0.60, . . . , 0.95}).* mAP_L, mAP_M *and* mAP_S *denote mAP of objects with* $area < 32^2$, $area > 32^2 < 96^2$ *and* $area > 96^2$ *respectively.*" [69]. This table is adapted from [69], Paper F

The evaluation of the proposed method is two-fold: Firstly the overall performance in mAP is calculated across the entire dataset (shown in Table 4.9) as well as the mAE and St.Dev. of the prediction head (shown in Table 4.10). Secondly, the performance is also calculated with respect to object size, to investigate if any categories are neglected to reach a more generalized solution. Any correlations between object size and weather might be a result of the latent representation inadvertently favoring denser parts of the size distribution, particularly small objects. Additionally due to the performance observed in [72], MR is also reported, as context awareness might allow the models to recognize objects missed by the baseline.

As can be seen in Table 4.9, the conditioned models generally underperform their respective baselines. Notably, the performance of temperature and

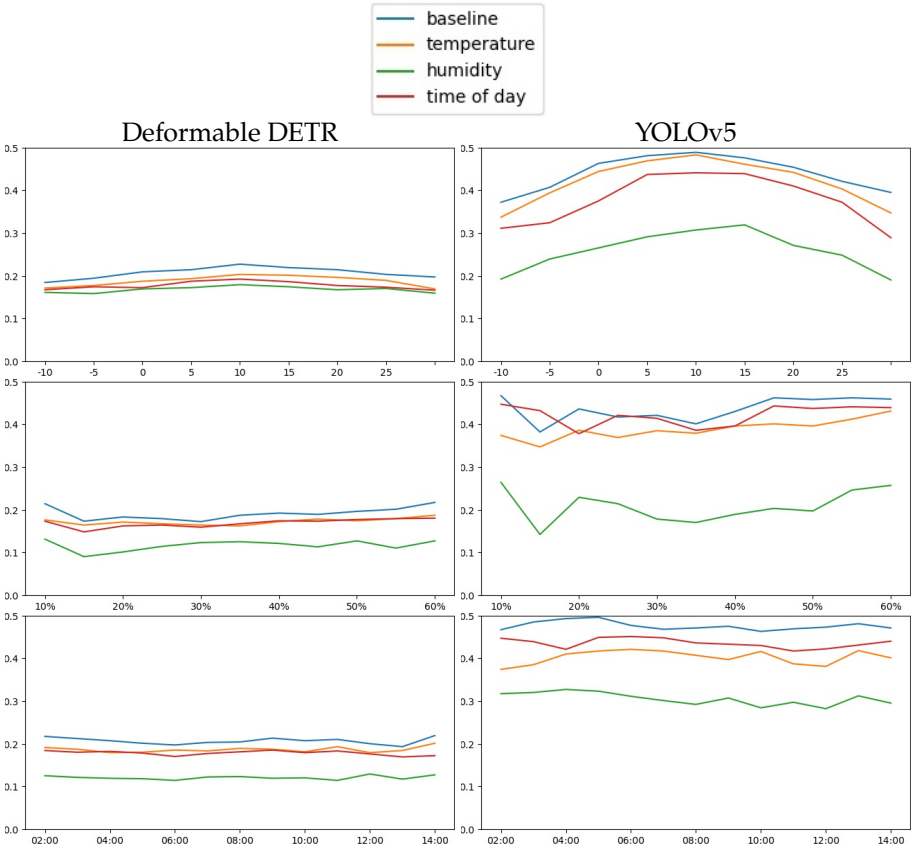


Fig. 4.10: This figure visualizes $mAP_{0.5}$ with respect to temperature (top row), humidity (middle row), and time of day (bottom row). This figure was adapted from [69] (Paper F)

time-of-day variants are only slightly lower than baselines, whereas humidity-conditioned approaches are significantly lower than baselines for the direct conditioning models. In Table 4.10 it can be seen that the accuracy of the weather-predictive branches also follows a similar pattern, and generally display difficulty in correctly predicting their respective weather conditions. Interestingly predicting temperature displays an average error slightly higher than the allowed deviance the St.Dev. is rather low, indicating that it is generally quite accurate at predicting close to the right temperature, however when it fails, it fails quite drastically.

While it has been shown that these weather signals can be extracted in traditional weather classification [22, 26, 81], it seems that extracting such a signal is exceedingly difficult in the thermal domain. W Notably the only improvement is seen with temperature-conditioned models, which display an

improvement in terms of MR. Intuitively, during concept drift, artifacts may appear suitable for an object in one distribution but undesired in another. During training, the model may adjust to either over-predict (increased false positives) or under-predict (increased false negatives) when concept drift occurs. Intuitively, conditioning on underlying drift components should have allowed the network to reason about the underlying data distribution and thus learn robust patterns.

While the mAP scores in Table 4.9 do not improve over the baseline when conditioned with the auxiliary branch, there is an indication that the auxiliary branch enforces a signal related to the auxiliary task. Particularly, the temperature-conditioned variant successfully detects objects that the baseline fails to detect but in turn, leads to an increase in False Positive (FP). Furthermore, the transformer-based model exhibits more uniform performance across temperatures, which may be due to the auxiliary predictive branch or the nature of transformer input-dependent attention. Surprisingly, DETR-variants perform well on small objects, contrary to previous observations [15, 79, 158].

	Auxiliary Prediction Accuracy		
	Model	mAE	St.Dev.
Dir.	Temperature	7.1	3.7
	Humidity	18.9	9.4
	Time of Day	7.3	7.1
Indir.	Temperature	5.1	2.9
	Humidity	15.3	8.9
	Time of Day	8.3	7.9

Table 4.10: *"Accuracy of the predicted auxiliary prediction value, Dir. and Indir. denotes the direct- and indirect-conditioning models respectively, while the model row denotes the variant used."* [69]. This table is adapted from [69] Paper F

Perhaps extracting weather conditions from a relative-thermal camera is ill-suited for precise regression. As day/night binary classification has been proven effective, future work could attempt varying granularities of binning to identify the ideal with accuracy that can be accurately modeled.

5.4 Summary and Contributions

Thermal concept drift poses a challenging hurdle to overcome during the deployment of thermal object detection tasks. Taking inspiration from related work [97, 108, 123, 138] on context awareness a study into leveraging granular meta-data to infuse weather awareness into models was conducted. Leveraging both direct and indirect conditioning methods an attempt to accurately predict weather conditions as an auxiliary task was employed to condition intermediate representations. Regrettably, extracting an accurate weather signal from relative-thermal cameras as an auxiliary optimization goal proves ineffective. Guiding the model to directly infer granular predictions from relative thermal images could prove unnecessarily noisy.

In this section, the work conducted in Paper F was discussed. In large parts, the work focused on infusing intermediate representations with weather-related information through two conditioning methods, guided by an auxiliary

classification head. With the aim to obtain more robust features, the impact is measured with respect to known concept drift indicators The contributions of the work described herein can be summarized as follows:

- An analysis of direct- and indirect-conditioning of weather-related data for auxiliary guidance of Transformer and CNN-style object detectors.
- A detailed performance comparison of conditioning methods with respect to relevant concept drift indicators. Showing that while the impact is measurable in terms of performance, guidance with a granular weather prediction does not provide a clear enough signal to improve performance.

References

[1] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1367–1376, 2020.

[2] K. Akshatha, A. K. Karunakar, S. B. Shenoy, A. K. Pai, N. H. Nagaraj, and S. S. Rohatgi, "Human detection in aerial thermal images using faster r-cnn and ssd algorithms," *Electronics*, vol. 11, no. 7, p. 1151, 2022.

[3] M. Ball and H. Pinkerton, "Factors affecting the accuracy of thermal imaging cameras in volcanology," *Journal of Geophysical Research: Solid Earth*, vol. 111, no. B11, 2006.

[4] G. Batchuluun, J. K. Kang, D. T. Nguyen, T. D. Pham, M. Arsalan, and K. R. Park, "Deep learning-based thermal image reconstruction and object detection," *IEEE Access*, vol. 9, pp. 5951–5971, 2020.

[5] F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," *Knowledge-Based Systems*, vol. 245, p. 108632, 2022.

[6] A. Berg, J. Ahlberg, and M. Felsberg, "A thermal object tracking benchmark," in *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2015, pp. 1–6.

[7] E. Bernard, N. Rivière, M. Renaudat, M. Péalat, and E. Zenou, "Active and thermal imaging performance under bad weather conditions," 2014.

[8] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 682–11 692.

[9] A. Bobu, E. Tzeng, J. Hoffman, and T. Darrell, "Adapting to continuously shifting domains," 2018.

[10] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS - improving object detection with one line of code," in *ICCV*, 2017.

[11] I. Bozcan and E. Kayacan, "Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8504–8510.

[12] D. Bulanon, T. Burks, and V. Alchanatis, "Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection," *Biosystems Engineering*, vol. 101, no. 2, pp. 161–171, 2008.

[13] A. V. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.

[14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018.

[15] 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: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[16] S. S. Chaturvedi, L. Zhang, and X. Yuan, "Pay" attention" to adverse weather: Weather-aware attention-based object detection," in *2022 26th International Conference on Pattern Recognition (ICPR)*. IEEE, 2022, pp. 4573–4579.

[17] Y.-T. Chen, J. Shi, Z. Ye, C. Mertz, D. Ramanan, and S. Kong, "Multimodal object detection via probabilistic ensembling," in *European Conference on Computer Vision*. Springer, 2022, pp. 139–158.

[18] Y.-Y. Chen, S.-Y. Jhong, G.-Y. Li, and P.-H. Chen, "Thermal-based pedestrian detection using faster r-cnn and region decomposition branch," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2019, pp. 1–2.

[19] G. Cheng, X. Yuan, X. Yao, K. Yan, Q. Zeng, X. Xie, and J. Han, "Towards large-scale small object detection: Survey and benchmarks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[20] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[21] S. S. Chouhan, U. P. Singh, and S. Jain, "Applications of computer vision in plant pathology: a survey," *Archives of computational methods in engineering*, vol. 27, pp. 611–632, 2020.

[22] W.-T. Chu, X.-Y. Zheng, and D.-S. Ding, "Camera as weather sensor: Estimating weather information from single images," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 233–249, 2017.

[23] J. CORNÉ and U. H. SJÖBLOM, "Investigation of ir transmittance in different weather conditions and simulation of passive ir imaging for flight scenarios," Ph.D. dissertation, MS thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2019.

[24] K. Corona, K. Osterdahl, R. Collins, and A. Hoogs, "Meva: A large-scale multiview, multimodal video dataset for activity detection," in *Proceedings of the*

IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 1060–1068.

[25] ——, "Meva: A large-scale multiview, multimodal video dataset for activity detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1060–1068.

[26] K. Dahmane, P. Duthon, F. Bernardin, M. Colomb, F. Chausse, and C. Blanc, "Weathereye-proposal of an algorithm able to classify weather conditions from traffic camera images," *Atmosphere*, vol. 12, no. 6, p. 717, 2021.

[27] R. Dai, M. Lefort, F. Armetta, M. Guillermin, and S. Duffner, "Self-supervised continual learning for object recognition in image sequences," in *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*. Springer, 2021, pp. 239–247.

[28] X. Dai, X. Yuan, and X. Wei, "Tirnet: Object detection in thermal infrared images for autonomous driving," *Applied Intelligence*, vol. 51, no. 3, pp. 1244–1261, 2021.

[29] D. C. De Oliveira and M. A. Wehrmeister, "Using deep learning and low-cost rgb and thermal cameras to detect pedestrians in aerial images captured by multirotor uav," *Sensors*, vol. 18, no. 7, p. 2244, 2018.

[30] C. Devaguptapu, N. Akolekar, M. M Sharma, and V. N Balasubramanian, "Borrow from anywhere: Pseudo multi-modal object detection in thermal imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[32] S. Disabato and M. Roveri, "Learning convolutional neural networks in presence of concept drift," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[33] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2011.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[35] N. M. K. Dousai and S. Lončarić, "Detecting humans in search and rescue operations based on ensemble learning," *IEEE Access*, vol. 10, pp. 26 481–26 492, 2022.

[36] Y. Dèveloppement, "Thermal imagers and detectors 2020 - covid-19 outbreak impact – preliminary report," http://www.yole.fr/Thermal_Imagers_And_Detectors_Covid19_Outbreak_Impact.aspx, 2020, accessed: 2021-08-11.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, pp. 303–338, 2010.

[38] D. M. Farid, L. Zhang, A. Hossain, C. M. Rahman, R. Strachan, G. Sexton, and K. Dahal, "An adaptive ensemble classifier for mining concept drifting data streams," *Expert Systems with Applications*, vol. 40, no. 15, pp. 5895–5906, 2013.

[39] W. Filho, "Distance correlation," https://gist.github.com/wladston, 2020, accessed: 2021-07-22.

[40] FLIR, "Qualitative vs. quantitative thermography: Understanding what is required and when," https://www.flir.com/discover/professional-tools/qualitative-vs.-quantitative-thermography-understanding-what-is-required-and-when/, accessed: 2023-08-23.

[41] R. Gade and T. B. Moeslund, "Thermal cameras and applications: a survey," *Machine vision and applications*, vol. 25, pp. 245–262, 2014.

[42] H. Gajjar, S. Sanyal, and M. Shah, "A comprehensive study on lane detecting autonomous car using computer vision," *Expert Systems with Applications*, p. 120929, 2023.

[43] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[44] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple Copy-Paste is a strong data augmentation method for instance segmentation," in *CVPR*, 2021.

[45] D. Ghose, S. M. Desai, S. Bhattacharya, D. Chakraborty, M. Fiterau, and T. Rahman, "Pedestrian detection in thermal images using saliency maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[46] J. F. S. Gomes and F. R. Leta, "Applications of computer vision techniques in the agriculture and food industry: a review," *European Food Research and Technology*, vol. 235, pp. 989–1000, 2012.

[47] V. P. Goncalves, L. P. Silva, F. L. Nunes, J. E. Ferreira, and L. V. Araújo, "Concept drift adaptation in video surveillance: a systematic review," *Multimedia Tools and Applications*, pp. 1–41, 2023.

[48] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López, "Pedestrian detection at day/night time with visible and fir cameras: A comparison," *Sensors*, vol. 16, no. 6, p. 820, 2016.

[49] A. Gowen, B. Tiwari, P. Cullen, K. McDonnell, and C. O'Donnell, "Applications of thermal imaging in food quality and safety assessment," *Trends in food science & technology*, vol. 21, no. 4, pp. 190–200, 2010.

[50] Ö. Gözüaçık and F. Can, "Concept learning using one-class classifiers for implicit drift detection in evolving data streams," *Artificial Intelligence Review*, vol. 54, pp. 3725–3747, 2021.

[51] T. Gruber, F. Julca-Aguilar, M. Bijelic, and F. Heide, "Gated2depth: Real-time dense lidar from gated images," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[52] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 305–310.

[53] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-NMS for video object detection," *CoRR*, vol. abs/1602.08465, 2016.

[54] M. A. Hashmani, S. M. Jameel, H. Al-Hussain, M. Rehman, and A. Budiman, "Accuracy performance degradation in image classification models due to concept drift," *Int. J. Adv. Comput. Sci. Appl*, vol. 10, 2019.

[55] Y. He, B. Deng, H. Wang, L. Cheng, K. Zhou, S. Cai, and F. Ciampa, "Infrared machine vision and infrared thermography with deep learning: A review," *Infrared physics & technology*, vol. 116, p. 103754, 2021.

[56] Hikvision, "Ds-2td2235d-25/50," https://us.hikvision.com/en/products/more-products/discontinued-products/thermal-camera/thermal-network-bullet-camera-ds, 2015, accessed: 2021-05-27.

[57] S.-C. Huang, T.-H. Le, and D.-W. Jaw, "Dsnet: Joint semantic learning for object detection in inclement weather conditions," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 8, pp. 2623–2633, 2020.

[58] N. U. Huda, B. D. Hansen, R. Gade, and T. B. Moeslund, "The effect of a diverse dataset for transfer learning in thermal person detection," *Sensors*, vol. 20, no. 7, p. 1982, 2020.

[59] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baselines," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[60] D. M. Institute, "Dmi api," https://confluence.govcloud.dk/display/FDAPI, 2019, accessed: 2021-05-27.

[61] M. Intelligence, "Ir camera market - growth, trends, covid-19 impact, and forecasts (2021 - 2026)," https://www.mordorintelligence.com/industry-reports/ir-camera-market, 2021, accessed: 2021-08-11.

[62] R. Ippalapally, S. H. Mudumba, M. Adkay, and N. V. HR, "Object detection using thermal imagingd," in *2020 IEEE 17th India Council International Conference (INDICON)*. IEEE, 2020, pp. 1–6.

[63] C. James, A. Richardson, P. Watt, and N. Maxwell, "Reliability and validity of skin temperature measurement by telemetry thermistors and a thermal camera during exercise in the heat," *Journal of thermal biology*, vol. 45, pp. 141–149, 2014.

[64] M. Jaworski, L. Rutkowski, and P. Angelov, "Concept drift detection using autoencoders in data streams processing," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2020, pp. 124–133.

[65] A. Jiang, R. Noguchi, and T. Ahamed, "Tree trunk recognition in orchard autonomous operations under different light conditions using a thermal camera and faster r-cnn," *Sensors*, vol. 22, no. 5, p. 2065, 2022.

[66] D. Jiang, D. Zhuang, Y. Huang, and J. Fu, "Survey of multispectral image fusion techniques in remote sensing applications," in *Image fusion and its applications*. IntechOpen, 2011, vol. 1, pp. 1–22.

[67] D. M. Jiménez-Bravo, P. M. Mutombo, B. Braem, and J. M. Marquez-Barja, "Applying faster r-cnn in extremely low-resolution thermal images for people detection," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2020, pp. 1–4.

[68] A. S. Johansen, J. C. J. Junior, K. Nasrollahi, S. Escalera, and T. B. Moeslund, "Chalearn lap seasons in drift challenge: Dataset, design and results," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Springer, 2022, pp. 755–769.

[69] A. S. Johansen, K. Nasrollahi, S. Escalera, and T. B. Moeslund, "Who cares about the weather?: Inferring weather conditions for weather-aware object detection in thermal images," in *Applied Sciences: New Trends on Pattern Recognition and Computer Vision, Applications and Systems*. MDPI, 2023.

[70] R. Kalsotra and S. Arora, "Background subtraction for moving object detection: explorations of recent developments and challenges," *The Visual Computer*, vol. 38, no. 12, pp. 4151–4178, 2022.

[71] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *CVPR-W*, 2016.

[72] M. Kieu, A. D. Bagdanov, M. Bertini, and A. Del Bimbo, "Task-conditioned domain adaptation for pedestrian detection in thermal imagery," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 546–562.

[73] J. Kim, H. Kim, T. Kim, N. Kim, and Y. Choi, "Mlpd: Multi-label pedestrian detector in multispectral domain," *Robotics and Automation Letters*, vol. 6, no. 4, 2021.

[74] A. H. Ko, R. Sabourin, and A. S. Britto Jr, "From dynamic classifier selection to dynamic ensemble selection," *Pattern recognition*, vol. 41, no. 5, pp. 1718–1731, 2008.

[75] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, "Weighted ensemble object detection with optimized coefficients for remote sensing images," *ISPRS International Journal of Geo-Information*, vol. 9, no. 6, p. 370, 2020.

[76] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE access*, vol. 8, pp. 125 459–125 476, 2020.

[77] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajník, "Artificial intelligence for long-term robot autonomy: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4023–4030, 2018.

[78] Z. Kütük and G. Algan, "Semantic segmentation for thermal images: A comparative survey," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 286–295.

[79] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 619–13 627.

[80] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "Cbnetv2: A composite backbone network architecture for object detection," *CoRR*, vol. abs/2107.00420, 2021.

[81] D. Lin, C. Lu, H. Huang, and J. Jia, "Rscm: Region selection and concurrency model for multi-class weather recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4154–4167, 2017.

[82] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[83] M. Liu, J. Jiang, C. Zhu, and X.-C. Yin, "Vlpd: Context-aware pedestrian detection via vision-language semantic self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6662–6671.

[84] Q. Liu, Z. He, X. Li, and Y. Zheng, "Ptb-tir: A thermal infrared pedestrian tracking benchmark," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 666–675, 2019.

[85] W. Liu, D. L. W. Luo, and S. Gao, "Future frame prediction for anomaly detection – a new baseline," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[86] W. Liu, G. Ren, R. Yu, S. Guo, J. Zhu, and L. Zhang, "Image-adaptive yolo for object detection in adverse weather conditions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 2, 2022, pp. 1792–1800.

[87] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[88] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2720–2727.

[89] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[90] B. I. F. Maciel, S. G. T. C. Santos, and R. S. M. Barros, "A lightweight concept drift detection ensemble," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2015, pp. 1061–1068.

[91] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[92] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1975–1981.

[93] A. McKinney, "Vqvae2 implementation," 2021, last accessed: June 7, 2021. [Online]. Available: https://github.com/vvvm23/vqvae-2

[94] C. Mera, M. Orozco-Alzate, and J. Branch, "Incremental learning of concept drift in multiple instance learning for industrial visual inspection," *Computers in Industry*, vol. 109, pp. 153–164, 2019.

[95] M. J. Mirza, C. Buerkle, J. Jarquin, M. Opitz, F. Oboril, K.-U. Scholl, and H. Bischof, "Robustness of object detectors in degrading weather conditions," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 2719–2724.

[96] U. Mittal, P. Chawla, and R. Tiwari, "Ensemblenet: A hybrid approach for vehicle detection and estimation of traffic density based on faster r-cnn and yolo models," *Neural Computing and Applications*, vol. 35, no. 6, pp. 4755–4774, 2023.

[97] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1396–1404.

[98] F. Munir, S. Azam, and M. Jeon, "Sstn: Self-supervised domain adaptation thermal object detection for autonomous driving," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 206–213.

[99] C. Murdock, Z. Li, H. Zhou, and T. Duerig, "Blockout: Dynamic model selection for hierarchical deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2583–2591.

[100] P. Nagar, M. Khemka, and C. Arora, "Concept drift detection for multivariate data streams and temporal segmentation of daylong egocentric videos," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1065–1074.

[101] J. Nataprawira, Y. Gu, I. Goncharenko, and S. Kamijo, "Pedestrian detection on multispectral images in different lighting conditions," in *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE, 2021, pp. 1–5.

[102] I. A. Nikolov, M. P. Philipsen, J. Liu, J. V. Dueholm, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Seasons in drift: A long-term thermal imaging dataset for studying concept drift," in *Thirty-fifth Conference on Neural Information Processing Systems*. Neural Information Processing Systems Foundation, 2021.

[103] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9725–9734.

[104] H. Y. Oh, M. S. Khan, S. B. Jeon, and M.-H. Jeong, "Automated detection of greenhouse structures using cascade mask r-cnn," *Applied Sciences*, vol. 12, no. 11, p. 5553, 2022.

[105] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*. IEEE, 2011, pp. 3153–3160.

[106] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 372–14 381.

[107] L. Pei, X. Yuan, and X. Dai, "Mwnet: object detection network applicable for different weather conditions," *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1394–1400, 2019.

[108] J. Peng, H. Wang, S. Yue, and Z. Zhang, "Context-aware co-supervision for accurate object detection," *Pattern Recognition*, vol. 121, p. 108199, 2022.

[109] J. Portmann, S. Lynen, M. Chli, and R. Siegwart, "People detection and tracking from aerial thermal views," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 1794–1800.

[110] M. Pranav, L. Zhenggang *et al.*, "A day on campus-an anomaly detection dataset for events in a single camera," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[111] B. Ramachandra and M. Jones, "Street scene: A new dataset and evaluation protocol for video anomaly detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2569–2578.

[112] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[113] C. J. Reed, X. Yue, A. Nrusimha, S. Ebrahimi, V. Vijaykumar, R. Mao, B. Li, S. Zhang, D. Guillory, S. Metzger *et al.*, "Self-supervised pretraining improves self-supervised pretraining," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2584–2594.

[114] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7374–7383.

[115] C. Sakaridis, D. Dai, and L. Van Gool, "Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding," *arXiv preprint arXiv:2104.13395*, 2021.

[116] K. K. Santhosh, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 6, pp. 1–26, 2020.

[117] D. M. V. Sato, S. C. De Freitas, J. P. Barddal, and E. E. Scalabrin, "A survey on concept drift in process mining," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–38, 2021.

[118] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[119] M. Schutera, F. M. Hafner, J. Abhau, V. Hagenmeyer, R. Mikut, and M. Reischl, "Cuepervision: self-supervised learning for continuous domain adaptation without catastrophic forgetting," *Image and Vision Computing*, vol. 106, p. 104079, 2021.

[120] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[121] J. Selva, A. S. Johansen, S. Escalera, K. Nasrollahi, T. B. Moeslund, and A. Clapés, "Video transformers: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[122] D. Shanmugam, D. Blalock, G. Balakrishnan, and J. Guttag, "Better aggregation in test-time augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 1214–1223.

[123] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 4156–4166.

[124] P. Siva and T. Xiang, "Weakly supervised object detector learning with model drift detection," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 343–350.

[125] W. N. Street and Y. Kim, "A streaming ensemble algorithm (sea) for large-scale classification," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 377–382.

[126] N. Sugianto, D. Tjondronegoro, G. Sorwar, P. Chakraborty, and E. I. Yuwono, "Continuous learning without forgetting for person re-identification," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2019, pp. 1–8.

[127] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[128] G. J. Székely, M. L. Rizzo, N. K. Bakirov *et al.*, "Measuring and testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[129] H. Tang, Y. Zhao, and H. Lu, "Unsupervised person re-identification with iterative self-supervised domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[130] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, "Neural granger causality," *arXiv preprint arXiv:1802.05842*, 2018.

[131] B. Tejedor, E. Lucchi, and I. Nardi, "Application of qualitative and quantitative infrared thermography at urban level: Potential and limitations," in *New Technologies in Building and Construction: Towards Sustainable Development*. Springer, 2022, pp. 3–19.

[132] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton, "Computer vision for sports: Current applications and research topics," *Computer Vision and Image Understanding*, vol. 159, pp. 3–18, 2017.

[133] X. Tian, W. W. Ng, and H. Xu, "Deep incremental hashing for semantic image retrieval with concept drift," *IEEE Transactions on Big Data*, 2023.

[134] Q. H. Tran, D. Han, C. Kang, A. Haldar, and J. Huh, "Effects of ambient temperature and relative humidity on subsurface defect detection in concrete structures by active thermal imaging," *Sensors*, vol. 17, no. 8, p. 1718, 2017.

[135] Ultralytics, "Yolov5," 2020, last accessed: April 15, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[136] J. Vertens, J. Zürn, and W. Burgard, "Heatnet: Bridging the day-night domain gap in semantic segmentation with thermal images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8461–8468.

[137] V. Vs, D. Poster, S. You, S. Hu, and V. M. Patel, "Meta-uda: Unsupervised domain adaptive thermal object detection using meta-learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 1412–1423.

[138] A. Wang, Y. Sun, A. Kortylewski, and A. L. Yuille, "Robust object detection under occlusion with context-aware compositionalnets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 645–12 654.

[139] C. Wang, A. Bochkovskiy, and H. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *CVPR*, 2021.

[140] L. Wang, H. Qin, X. Zhou, X. Lu, and F. Zhang, "R-yolo: A robust object detector in adverse weather," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–11, 2022.

[141] D. Weiss, B. Sapp, and B. Taskar, "Dynamic structured model selection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2656–2663.

[142] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 965–23 998.

[143] M. Wulfmeier, A. Bewley, and I. Posner, "Incremental adversarial domain adaptation for continually changing environments," in *2018 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4489–4495.

[144] A. I. Xavier, C. Villavicencio, J. J. Macrohon, J.-H. Jeng, and J.-G. Hsieh, "Object detection via gradient-based mask r-cnn using machine learning algorithms," *Machines*, vol. 10, no. 5, p. 340, 2022.

[145] J. Xie, Y. Zheng, R. Du, W. Xiong, Y. Cao, Z. Ma, D. Cao, and J. Guo, "Deep learning-based computer vision for surveillance in its: Evaluation of state-of-the-art methods," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3027–3042, 2021.

[146] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, vol. 7, pp. 156 694–156 706, 2019.

[147] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "{CADE}: Detecting and explaining concept drift samples for security applications," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[148] S. Ye, Y. Shi, R. Wang, Y. Wang, J. Xu, C. Yang, and X. You, "Cdlt: A dataset with concept drift and long-tailed distribution for fine-grained visual categorization," *arXiv preprint arXiv:2306.02346*, 2023.

[149] A. Yeshchenko, C. Di Ciccio, J. Mendling, and A. Polyvyanyy, "Comprehensive process drift detection with visual analytics," in *International Conference on Conceptual Modeling*. Springer, 2019, pp. 119–135.

[150] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[151] C. Yuan and S. S. Agaian, "Bithermalnet: A lightweight network with bnn rpn for thermal object detection," in *Multimodal Image Exploitation and Learning 2022*, vol. 12100. SPIE, 2022, pp. 114–123.

[152] F. Zeng, Y. Ji, and M. D. Levine, "Contextual bag-of-words for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1433–1447, 2017.

[153] H. Zhang, X.-g. Hong, and L. Zhu, "Detecting small objects in thermal images using single-shot detector," *Automatic Control and Computer Sciences*, vol. 55, no. 2, pp. 202–211, 2021.

[154] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon, "Guided attentive feature fusion for multispectral pedestrian detection," in *WACV*, 2021.

[155] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *CoRR*, vol. abs/1710.09412, 2017.

[156] Y. Zheng, I. H. Izzat, and S. Ziaee, "Gfd-ssd: gated fusion double ssd for multispectral pedestrian detection," *arXiv preprint arXiv:1903.06999*, 2019.

[157] K. Zhou, L. Chen, and X. Cao, "Improving multispectral pedestrian detection by addressing modality imbalance problems," *arXiv preprint arXiv:2008.03043*, 2020.

[158] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

[159] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," *Big data analysis: new algorithms for a new society*, pp. 91–114, 2016.

[160] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

Chapter 5

Conclusion

This PhD Thesis was conducted in collaboration with Milestone Systems A/S as a part of the Milestone Research Programme at Aalborg University in the period from 2020 to 2023. The thesis focused on investigating generic object detection and segmentation methods for image understanding in real-world contexts. Encompassing frameworks for joint learning of semantic segmentation and super-resolution, an extensive study on the use of transformers in the video domain, and an in-depth analysis of the impact of concept drift on thermal object detection as well as introducing a novel dataset for future research on the topic of thermal concept drift.

The main contributions of this PhD can be summarized as follows:

- Introduction of novel multi-task frameworks and techniques which improve the performance of semantic segmentation and super-resolution algorithms for real-world applications.
- State-of-the-art accuracy on the challenging Cityscapes dataset with 80.3% and 79.0% on the validation- and test-set respectively.
- State-of-the-art accuracy on the challenging IDD-Lite dataset with 76.3% on the validation set.
- A detailed survey of trends, techniques, and challenges faced when employing transformers to model video data. Notably, identifying current shortcomings, pitfalls, and promising techniques for designing efficient video transformers.
- The development of the Long-Term Drift (LTD) dataset, introducing the largest public dataset for studying thermal concept drift in stationary setups. Consisting of more than 298 hours of video with rich meta-data for both weather conditions and temporal notation.

- An in-depth analysis of the performance impact induced by thermal concept drift, identifying correlated weather conditions and conducting a fine-grained performance analysis with respect to said conditions.
- Investigated multiple conditioning methods and architectures for learning weather-aware latent representations for robust object detection during long-term deployment and underlining the difficulty of extracting a weather signal from relative thermal cameras.

In conclusion, we explored various novel approaches and frameworks aimed at enhancing the capabilities of computer vision systems for real-world applications. Development of novel frameworks such as Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR) demonstrated the remarkable improvements to segmentation models that can be obtained by jointly learning super-resolution and semantic segmentation. Additionally, the integration of semantic guidance into real-world super-resolution, as shown with Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR), exemplifies that semantic guidance can further provide superior perceptual consistency and noisy reduction for the super-resolution task.

Furthermore, we explored the challenges and trends that present themselves for transformers stepping beyond single-frame analysis and into the video domain. Notably, the computational burden that comes with global reasoning and high-dimensional data. Reducing input size, restricting the receptive field, or aggregating information through highly refined representations proves to be efficient methods for addressing the computational burden, while also exhibiting strong capabilities for object-centric tasks.

Lastly, the introduction of the Long-Term Drift (LTD) dataset has enabled a fine-grained study of the impact of thermal concept drift on vision tasks. Further underlining the importance of considering external factors, such as weather, for long-term deployments of vision systems. The subsequent extension, of the LTD dataset, and public challenge fostered advancements in robust object-detection algorithms, through fine-grained analysis of the impact of weather conditions on object detectors. Lastly, efforts to infuse weather awareness into models to counter concept drift provided insights into conditioning methods, emphasizing the need for effective guidance mechanisms.

The contributions presented in this PhD thesis collectively furthered the understanding of the impact and methods employed for real-world applications of object detection and segmentation algorithms. Despite these advances, real-world applications of computer vision continue to prove a challenging topic, with many challenges to solve. While the contributions presented in this thesis have provided some techniques to alleviate this problem, there is still a long way to go before we can transition object detection and segmentation algorithms to real-world deployments without considering the impact of concept

drift. Frameworks such as MT-SSSR and SSG-RWSR indicate the potential of jointly learning complex visual tasks. Combined with a similar observation within transformer architectures hints at multi-task learning as a promising tool to obtain a more efficient and holistic understanding of visual data. Further incentivizing the development of multifunctional models capable of addressing multiple challenges simultaneously and laying the foundation for more versatile and adaptable computer vision systems. Moreover, the exploration of transformers for video analysis highlights the importance of leveraging the temporal cues, providing a foundation for future research to unlock the full potential of these architectures in understanding dynamic visual scenes. Finally, addressing the challenges of thermal concept drift demonstrates the need to integrate external contextual cues into computer vision systems. Particularly understanding the nature of drift factors to properly identify them, and developing methods that allow for the application of this knowledge via architectural changes or augmenting the training phase to effectively create robust computer vision algorithms.

Overall, these findings collectively emphasize the importance of holistic, adaptable, and contextually aware computer vision systems, offering a glimpse into the potential transformative directions that the field may take in the pursuit of more intelligent and reliable visual understanding.

In case you have questions, comments, or suggestions, please do not hesitate to contact me. You can find my contact details below.

Anders Skaarup Johansen
asjo@create.aau.dk
Rendsburggade 14
9000 Aalborg

List of Abbreviations

ANN	Artificial Neural Network. 4, 7
AP	Average Precission. 21
CNN	Convolutional Neural Network. 16, 25, 40, 44–46, 48–52, 63, 132, 133, 274
DC	Distance Correlation. 74
DISTS	Deep Image Structure and Texture Similarity. 27, 29
DL	Deep Learning. 15, 16, 20
DSN	Degredation Simulation Network. 133
FoV	Field-of-View. 62, 70
FP	False Positive. 92
FPS	Frames Per Second. 78
FR-IQA	Full-Reference Image Quality Assessment. 143
GAN	Generative Adversarial Network. 15, 118, 119, 132, 133
GT	Ground-Truth. 117, 121, 129–131, 133, 135, 138, 143, 144
HR	High-Resolution. 13–15, 19–22, 24–27, 29, 31, 46, 117–119, 121, 129–137, 142, 145
ImageNet	ImageNet Large-Scale Visual Recognition Challenge. 5
IoU	Intersection over Union. 80, 90, 266, 270
IR	Infrared Radiation. 62, 72

KAIST	KAIST Multispectral Pedestrian Detection. 262–264
LPIPS	Learned Perceptual Image Patch Similarity. 15, 27, 29, 137, 138
LR	Low-Resolution. 13, 15, 16, 19–22, 24–27, 31, 117–119, 121, 129–137, 142, 143, 145, 146
LSTM	Long-Short-Term-Memory. 39, 41
LTD	Long-Term Drift. v, 70, 71, 76–78, 82, 84, 86, 105, 106, 261, 262, 264–266, 269
mAE	mean Average Error. 90, 92, 269
mAP	mean Average Precission. 21, 22, 24, 74, 80, 90–92, 262, 266, 269–271, 274, 275
Meta-IQA	Meta Image-Quality-Assesment. 27, 28
MHSA	Multi-Head Self-Attention. 44
ML	Machine Learning. 4
MOR	Mean Opinion Rank. 27–29, 143, 144
MR	Miss-Rate. 67, 90, 92, 262, 266, 269, 270, 274, 275
MS COCO	MicroSoft: Common Objects in Context. 5, 78, 79
MSE	Mean Squared Error. 15, 74–76, 117, 119, 129, 132, 134
MT-SSSR	Multi-Task Semantic Segmentation and Super-Resolution. v, vii, 19–24, 106, 107, 117, 119–123
MTM	Masked-Token-Modeling. 50
NIMA	Neural Image Assessment. 27, 28
NLP	Natural Language Processing. 39, 41, 47, 52
NR-IQA	No-Reference Image Quality Assessment. 143
OHEM	Online Hard Example Mining. 119, 122
OpenImages	Google Open Images. 5
PC	Pearson's Correlation. 74
PSNR	Peak Signal-to-Noise Ratio. 15, 27, 29, 132
RMI	Region Mutual Information. 21, 118, 119, 122
RNN	Recurrent Neural Network. 39, 41
RRDB	Residual-in-Residual Dense Block. 118, 137
RWSR	Real-World Super-Resolution. 14–16, 19, 24–27, 31, 129–133, 135, 142, 146

SA	Self-Attention. 41, 42, 44–47, 49
SNR	Signal-to-Noise Ratio. 118
SotA	State-of-the-Art. 21, 24, 25, 27, 28, 30, 31, 41, 44, 48, 52, 72, 83, 84, 118, 121, 125, 129, 131–133, 139–143, 146, 262
SR	Super-Resolution. 13–16, 18–28, 30, 31, 117–123, 129–139, 141–143, 146
SS	Semantic Segmentation. 14–16, 18–24, 26, 27, 31, 117–119, 121, 125, 131, 133, 134, 136, 138, 142, 146
SSG-RWSR	Semantic Segmentation Guided Real-World Super-Resolution. v, vii, 25–31, 106, 107, 129–131, 135, 136, 142, 143, 145, 146
SSIM	Structural Similarity index. 15, 19, 27, 29
St.Dev.	Standard Deviation. 90–92, 269
ViT	Vision Transformer. 39–41, 44–47, 49, 50, 53
VT	Video Transformer. v, 40, 45, 47–50, 52, 53

Part II

Papers

Paper A

Single-Loss Multi-Task Learning For Improving Semantic Segmentation Using Super-Resolution

Andreas Aakerberg, Anders S. Johansen, Kamal Nasrollahi and Thomas B. Moeslund

The paper has been published in the
Computer Analysis of Images and Patterns - 19th International Conference (CAIP), Lecture Notes in Computer Science, vol. 13053, pp. 403–411, 2021.

The layout has been revised.

Abstract

We propose a novel means to improve the accuracy of semantic segmentation based on multi-task learning. More specifically, in our Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR) framework, we jointly train a super-resolution and semantic segmentation model in an end-to-end manner using the same task loss for both models. This allows us to optimize the super-resolution model towards producing images that are optimal for the segmentation task, rather than ones that are of high-fidelity. Simultaneously we adapt the segmentation model to better utilize the improved images and thereby improve the segmentation accuracy. We evaluate our approach on multiple public benchmark datasets, and our extensive experimental results show that our novel MT-SSSR framework outperforms other state-of-the-art approaches.

1 Introduction

Semantic Segmentation (SS) is a widely studied computer vision problem that helps scene understanding by assigning dense labels to all pixels in an image. SS has several applications in fields such as autonomous driving, robot sensing, and similar tasks that require a semantic understanding with pixel-level localization. The accuracy of SS is highly correlated with the spatial resolution of the input images [23]. This is particularly prominent for segmentation of small objects, where High-Resolution (HR) is essential to obtain a high accuracy [10]. However, obtaining HR image data is not always possible. One possible solution is therefore to upsample Low-Resolution (LR) images as a pre-processing step. This can be done with classical interpolation-based methods, such as bicubic interpolation, or with the more recent deep-learning based Super-Resolution (SR) methods. The latter has shown to be the most effective in terms of restoring HR details from LR images [8, 24]. Deep-learning based SR models are trained by minimizing the loss, typically Mean Squared Error (MSE) loss, between the reconstructed HR image and the Ground-Truth (GT). Hence, these methods require paired LR/HR images for training. However, in the case of improving another computer vision task, such as SS, the objective and subjective quality of the super-resolved image is not necessarily the best metrics. Therefore, we hypothesize that by only using the segmentation loss, it is possible to optimize the SR model jointly, to produce super-resolved images that result in improved segmentation accuracy.

In this paper, we therefore propose a novel framework named Multi-Task Semantic Segmentation and Super-Resolution (MT-SSSR), for joint learning of SS and super-resolution as seen in Fig. A.1. We use ESRGAN [24] and HRNet [22] respectively as SR and SS backbones, in our joint framework, and

rely on a single loss for learning both models, namely the loss of the SS task. We evaluate our method on two different publicly available datasets, and present new State-of-the-Art (SotA) results on both. In summary, the contributions of this paper are:

- A novel multi-task learning framework, which uses a single loss to improve the segmentation performance together with SR.
- Our method does not require LR/HR training image pairs for the SR model when jointly learning in the multi-task learning framework.
- We outperform SotA SS methods on the challenging CityScapes and IDD-Lite datasets by respectively 4.2% and 2.2%, compared to the best existing published results.

2 Related Work

Super-resolution: Dong *et al.* [8] proposed the first deep-learning-based method for SR, which successfully learned to perform non-linear mapping from LR to HR images. Since then, most successful SR methods have been based on convolutional neural networks. One of the SotA SR methods is ESRGAN [24], which uses a relativistic Generative Adversarial Network (GAN) with Residual-in-Residual Dense Blocks (RRDBs). Besides improving Signal-to-Noise Ratio (SNR), or the perceptual quality of images, SR can also be used to assist other computer vision tasks to achieve better accuracy [7, 14]. Recently, it has been shown that SR can improve optical character recognition accuracy by up to 15% [15] and object detection in satellite imagery by up to 30% [18].
Semantic Segmentation: A popular method to achieve SS is to use an encoder-decoder architecture [2, 4, 17] which encodes the input image to dense representational feature-maps and then decodes to regain spatial information [12, 27]. Eff-UNet [3] utilizes Efficientnet [20] as an encoder and UNet [17] as a decoder, to achieve SotA performance the IDD-Lite dataset [13]. DeeplabV3 [5] uses atrous convolutions and skip-connections for decoding. ERFNet [16] uses deconvolutional layers, combined with a non-bottleneck-1d layer to reduce computational cost. PSPNet [26] proposes a spatial pyramid pooling layer that gathers information by pooling over an increasingly smaller region of the image, then fusing those feature-maps with the original feature-map. Unlike the previously mentioned methods, HRNet [22], aims to retain as much of the resolution of the input image, by combining a HR branch with parallel LR branches to achieve representational information, and subsequently fusing the information from all branches before the final layer. Segmentation models are often optimized using cross-entropy loss, which is a per-pixel evaluation. In [28], Region Mutual Information (RMI) loss is proposed, which utilizes

neighboring pixels in a statistical approach, allowing the model to adjust the loss based on how difficult the prediction is, resulting in an overall improvement in accuracy [28].

Multi-Task Learning: Multi-task learning has proven to be effective for different computer vision problems, when multiple tasks need to be solved at once. By jointly learning multiple related tasks, the performance of the individual tasks can be further improved, compared to learning them separately. In [9], multi-task learning is used to jointly learn image segmentation and depth estimation. In [25] it is proposed to use two GANs for joint de-noising and SR. In [11] a network that can perform a selection of tasks with the same weights is proposed. This is done with task-specific feature modulation, and residual adapters to adjust the forward pass. The work most closely related to ours is DSRL proposed in [23], where multi-task learning is used to jointly learn SR and SS. As the main purpose of multi-task learning in [23] is to improve the encoder of the segmentation model, the SR is considered an auxiliary task that is removed at test time. A key difference in our approach is that we use our SR model to upsample the input images during both training and testing. Additionally, we use the segmentation loss for optimizing our SR model, while [23] uses MSE, which requires a HR ground truth version of the input images for supervised learning.

3 The Proposed Framework

While the use of SR has shown to improve the performance of other vision tasks, experiments show that traditional SR metrics cannot be used as full proxies to recover all the lost details [7]. We postulate that using traditional SR metrics as auxiliary loss for multi-task learning, serves to optimize the model on some implicit assumptions rather than a global optimum for the entire system. We therefore propose using the segmentation task's loss for improving the performance of the SR task as well.

The block diagram of our proposed method, MT-SSSR, is shown in Fig. A.1. By jointly training both models using the segmentation loss, we remove the need for LR/HR image pairs during training. This makes our method applicable to real-world applications where such data are not available. The SR backbone in our framework is built upon the RRDB generator from ESRGAN [24]. Hence, we do not perform traditional GAN training with ESRGAN, and instead replace all pixel and feature-based loss functions with our task loss. For SS it is vital to have a high spatial resolution to accurately segment the contents of an image. Hence, we chose HRNet [22] as the backbone architecture. Other than replacing the Online Hard Example Mining (OHEM) cross-entropy loss [19] with RMI loss, we do not modify the HRNet architecture further.

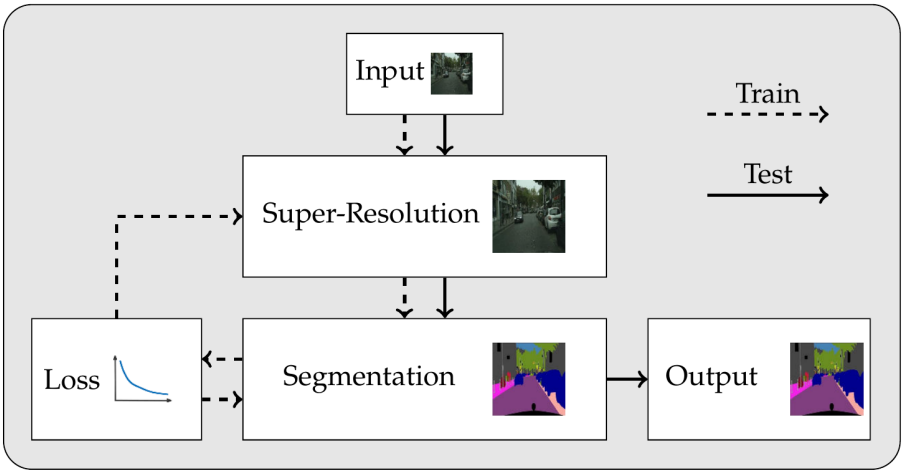


Fig. A.1: Our proposed framework, MT-SSSR. Dashed and full lines represent training and testing phases, respectively. The SR model learns to upsample and enhance the input image based on the segmentation task loss. The segmentation model uses the same loss to improve the accuracy of its prediction.

4 Experiments and Results

4.1 Datasets

The IDD dataset [21] contains driving scenes in unstructured environments, including both urban and rural scenes. In our experiments we use the IDD-Lite dataset [13] which is a sub-sampled version of the IDD dataset. The IDD-Lite dataset contains pixel annotations for 1404 training, 204 validation, and 408 test images, respectively. The dataset has a resolution of 320×227 pixels and contains 7 classes. Ground truth labels are only publicly available for the training and validation images.

The CityScapes dataset [6] contains driving scenes from 50 different cities recorded across several months. The dataset contains finely annotated semantic maps for 2975 training, 500 validation, and 1525 test images, respectively, which have a resolution of 2048×1024 pixels. Following [23], we sub-sample the CityScapes dataset to 1024×512 pixels. There are 19 classes to be segmented. We report our results on the test set, based on submission to the CityScapes Online Server.

4.2 Implementation Details

For both our experiments on CityScapes and IDD-Lite, we initialize the segmentation backbone with weights pre-trained on CityScapes training data. For

the SR backbone, we use transfer-learning by pre-training the model on generic LR/HR image pairs before the model is used in the multi-task framework. For this, we use the DF2K dataset, which is a merge of DIV2K [1] and Flickr2K [24], and use bicubic interpolation to downsample the HR images. We denote the pre-trained SR model as SR_{ST} (Super-Resolution$_{Single-Task}$).

For our experiments on CityScapes, we use the sub-sampled images, but test against the full-resolution labels by upsampling our predictions with bi-linear interpolation. For our experiments on IDD-Lite we train at the native resolution training images and labels, and test against 256×128 pixel labels according to [13]. We experiment with both $\times 2$ and $\times 4$ upsampling in our MT-SSSR framework.

Training Setup: Due to memory constraints, we use a cyclic approach for training our MT-SSSR framework, where we alternate between training on patches and the full image. For patch training, we randomly crop 128×128 pixel LR patches from the training images and update both the weights of the SR and the SS model. When training on the full image, we only update the weights of the SS model.

We train all our models using gradient-descent with a mini-batch size of 12 on four V100 GPUs using a learning rate of 0.001 with an exponential decay ($lr \times \frac{iter_{cur}}{iter_{max}}^{0.9}$) trained until convergence. For the segmentation models we additionally use momentum (0.9) and weight decay (0.0005).

4.3 Results

Results on CityScapes: Table A.1 shows the segmentation accuracy on CityScapes. We include results for experiments with 1024×512 resolution input images and $\times 2$ upsampling of these. Most noticeably, our MT-SSSR framework provides 4.2% improvement over the current SotA [23] and 3.6% improvement over the HRNet baseline [22] on the test set. As seen in the qualitative comparison in Fig. A.2, our jointly trained SR model enhances sharpness and details of the input images, which in turn helps the segmentation model to better segment smaller distant objects, compared to the baseline.

Results on IDD-Lite: The segmentation accuracy on IDD-Lite reported in Table A.3 shows that the performance increases with the upsampling factor in our MT-SSSR. In particular, our method with $\times 4$ upsampling provides 2.5% improvement compared to the current SotA [3] and 6.9% improvement over the baseline HRNet [22]. In the qualitative segmentation results in Fig. A.2, it can be seen that our method more accurately segments fine details in the image, compared to the baseline. This is also reflected in the per-class performance in Table A.2. An interesting example can be seen for the triangular part of the pole in the upper left corner (row three), where our method can label the sky correctly, even though this is mislabeled in the GT.

Method	Scale Factor	Val. (%)	Test (%)
DeepLabV3+ [5]	Native	70.0	67.1
PSPNet [26]	Native	71.5	69.1
HRNet [22]	Native	77.3	75.4
DSRL [23]	$\times 2$ SR_{MT}	75.7	74.8
MT-SSSR (ours)	$\times 2$ $\mathbf{SR}_{MT}$	**80.3**	**79.0**

Table A.1: Quantitative segmentation results on CityScapes.

Method	Drivable	Non Drivable	Living Things	Vehicles	Roadside Objects	Far Objects	Sky
HRNet + RMI	94.78	43.16	51.10	77.80	51.93	75.97	94.72
Eff-UNet [3]	94.86	**50.12**	61.96	81.31	54.99	77.54	95.55
MT-SSSR (ours best)	**95.07**	47.69	**68.50**	**85.97**	**59.01**	**80.91**	**96.66**

Table A.2: Per-class accuracy on IDD-Lite. Compared to the baseline and Eff-UNet our method improves significantly on small objects such as living things and roadside objects.

4.4 Ablation Study

Effect of Upsampling and RMI-loss: We investigate the effect of SR and RMI-loss on the CityScapes and IDD-Lite datasets. As seen in Table A.4 and A.5, RMI-loss improves slightly over using OHEM-loss in the baseline HRNet on both datasets. Furthermore, naively upsampling the input images with $\times 2$ bicubic interpolation also improves the performance slightly. However, when using $\times 4$ upsampling with bicubic interpolation on the IDD-Lite dataset, the accuracy drops 2.3% below baseline. Using images upsampled in a pre-processing step with the pre-trained single-task SR model, SR_{SR}, together with HRNet + RMI, provides 0.7 and 1.3% improvement over the baseline on CityScapes and IDD-Lite, respectively. When combining RMI-loss and SR in our multi-task framework we improve the performance by 3.6% and 6.9% for the CityScapes and IDD-Lite datasets, respectively.

Inference Time: We compare our method in terms of inference time against the baseline model on the CityScapes dataset, on a V100 GPU. The inference time is 101ms and 1888ms per image for the baseline HRNet and MT-SSSR, respectively. The inference time of HRNet at the increased resolution alone is 592ms per image. This means that the increased performance comes at a significant computational cost. However, no particular efforts has been made in order to optimize the inference time.

Method	Scale Factor	Val. (%)
DeepLabV3+ [5]	Native	64.3
ERFNet [16]	Native	66.1
HRNet [22]	Native	69.4
Eff-UNet [3]	Native	73.8
MT-SSSR (ours)	$\times 2$ SR_{MT}	74.1
MT-SSSR (ours)	$\times 4$ $\mathbf{SR}_{MT}$	**76.3**

Table A.3: Quantitative segmentation results on IDD-Lite.

Method	Scale Factor	Val. (%)
HRNet [22]	Native	77.3
HRNet + RMI	Native	77.4
HRNet + RMI	$\times 2$ Bicubic	78.0
HRNet + RMI	$\times 2$ SR_{ST}	78.1
MT-SSSR (ours)	$\times 2$ $\mathbf{SR}_{MT}$	**80.3**

Table A.4: The effect of RMI-loss and SR on segmentation accuracy on the CityScapes dataset. $_{MT}$ and $_{ST}$ denote multi-task and single-task, respectively.

Method	Scale Factor	Val. (%)
HRNet [22]	Native	69.4
HRNet + RMI	Native	69.9
HRNet + RMI	$\times 2$ Bicubic	70.9
HRNet + RMI	$\times 4$ Bicubic	67.1
HRNet + RMI	$\times 2$ SR_{ST}	71.2
MT-SSSR (ours)	$\times 2$ SR_{MT}	74.1
MT-SSSR (ours)	$\times 4$ $\mathbf{SR}_{MT}$	**76.3**

Table A.5: The effect of RMI-loss and SR on segmentation accuracy on the IDD-Lite dataset. $_{MT}$ and $_{ST}$ denote multi-task and single-task, respectively.

Fig. A.2: Comparison of segmentation results on CityScapes (rows 1,2) and IDD-Lite (row 3). The three first columns show the input image together with a zoomed-in crop of the input and super-resolved image respectively. The last three columns show the differences between the ground truth, HRNet [22] (baseline), and our best performing model respectively. Noticeable differences include; distant streetlights, poles and signs in row 1, traffic poles, and people in row 2, and distant poles and people in row 3.

5 Conclusion

In this paper, we propose a novel framework for SS based on multi-task learning with super-resolution. The super-resolution model learns to enhance the input images such that they become more suitable for the SS model, while the segmentation model jointly learns to predict more accurate segmentation maps. Our experimental results show that our proposed system outperforms existing SotA SS methods significantly on the challenging CityScapes and IDD-Lite datasets.

6 Acknowledgements

This work was partially supported by the Milestone Research Programme at Aalborg University (MRPA) and Danmarks Frie Forskningsfond (DFF 8022-00360B)

References

[1] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR*, 2017.

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *TPAMI*, 2017.

[3] B. Baheti, S. Innani, S. Gajre, and S. N. Talbar, "Eff-unet: A novel architecture for semantic segmentation in unstructured environment," in *CVPR*, 2020.

[4] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017.

[5] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[7] D. Dai, Y. Wang, Y. Chen, and L. Van Gool, "Is image super-resolution helpful for other vision tasks?" in *WACV*, 2016.

[8] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *TPAMI*, 2016.

[9] A. Jha, A. Kumar, S. Pande, B. Banerjee, and S. Chaudhuri, "MT-UNET: A novel u-net based multi-task architecture for visual scene understanding," in *ICIP*, 2020.

[10] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks," in *CVPR-W*, 2016.

[11] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, "Attentive single-tasking of multiple tasks," in *CVPR*, 2019.

[12] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *arXiv preprint*, 2020.

[13] A. Mishra, S. Kumar, T. Kalluri, G. Varma, A. Subramaian, M. Chandraker, and C. V. Jawahar, "Semantic segmentation datasets for resource constrained training," in *NCVPRIPG*, vol. 2, no. 3, 2020, p. 6.

[14] B. Na and G. C. Fox, "Object classifications by image super-resolution preprocessing for convolutional neural networks," *ASTESJ*, vol. 5, no. 2, pp. 476–483, 2020.

[15] V. Robert and H. Talbot, "Does super-resolution improve OCR performance in the real world? A case study on images of receipts," in *ICIP*, 2020.

[16] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *T-ITS*, vol. 19, no. 1, pp. 263–272, 2018.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICAI*, 2015, pp. 234–241.

[18] J. Shermeyer and A. Van Etten, "The effects of super-resolution on object detection performance in satellite imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[19] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *CVPR*, 2016.

[20] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *ICML*, 2019, pp. 6105–6114.

[21] G. Varma, A. Subramanian, A. M. Namboodiri, M. Chandraker, and C. V. Jawahar, "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments," in *WACV*, 2019.

[22] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2019.

[23] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in *CVPR*, 2020, pp. 3774–3783.

[24] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *ECCV*, 2019.

[25] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *CVPR*, 2018.

[26] H. Z., J. S., X. Q., X. W., and J. J., "Pyramid scene parsing network," in *CVPR*, 2017.

[27] B. Zhao, J. Feng, X. Wu, and S. Yan, "A survey on deep learning-based fine-grained object classification and semantic segmentation," *IJAC*, vol. 14, no. 2, pp. 119–135, 2017.

[28] S. Zhao, Y. Wang, Z. Yang, and D. Cai, "Region mutual information loss for semantic segmentation," in *NIPS*, 2019.

Paper B

Semantic Segmentation Guided Real-World Super-Resolution

Andreas Aakerberg, Anders S. Johansen, Kamal Nasrollahi, Thomas B. Moeslund

The paper has been published in the
IEEE/CVF Winter Conference on Applications of Computer Vision Workshops, WACV - Workshops, pp. 449–458, 2022.

The layout has been revised.

Abstract

Real-world single image Super-Resolution (SR) aims to enhance the resolution and reconstruct High-Resolution (HR) details of real Low-Resolution (LR) images. This is different from the traditional SR setting, where the LR images are synthetically created, typically with bicubic downsampling. As the degradation process for real-world LR images are highly complex, SR of such images is much more challenging. Recent promising approaches to solve the Real-World Super-Resolution (RWSR) problem include the use of domain adaptation to create realistic training-pairs, and self-learning based methods which learn an image specific SR model at test time. However, as domain adaptation is an inherently challenging problem in itself, SR models based solely on this approach are limited by the domain gap. In contrast, while self-learning based methods remove the need for paired-training data by utilizing internal information in the LR image, these methods come with the cost of slow prediction times. This paper proposes a novel framework, Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR), which uses an auxiliary semantic segmentation network to guide the SR learning. This results in noise-free reconstructions with accurate object boundaries, and enables training on real LR images. The latter allows our SR network to adapt to the image specific degradations, without Ground-Truth (GT) reference images. We support the guidance with domain adaptation to faithfully reconstruct realistic textures, and ensure color consistency. We evaluate our proposed method on two public available datasets, and present State-of-the-Art results in terms of perceptual image quality on both real and synthesized LR images.

1 Introduction

Single image Super-Resolution (SR) aims to upsample a Low-Resolution (LR) image and reconstruct the missing high-frequency details. SR has been a widely studied problem for decades, due to its vast number of applications in fields such as medical imaging, remote sensing, and surveillance. In latter, SR are often used to improve the performance of down-stream vision tasks, such as object detection and tracking, by improving the visibility of the images which often suffer from low-resolution due to the wide field-of-view and large object to camera distance. Traditionally, most work has been focusing on improving the fidelity of the images by minimizing the Mean Squared Error (MSE). However, recently more focus has been put into generating realistic High-Resolution (HR) images as perceived by humans [20]. Current State-of-the-Art (SotA) deep learning-based SR methods most often require paired LR/HR images to be trained by supervised learning. Commonly, researchers have been using artificial LR images created by downsampling HR images, typically using bicubic interpolation. However, this strategy changes the natural image characteristics, such as sensor noise and other corruptions, which limits

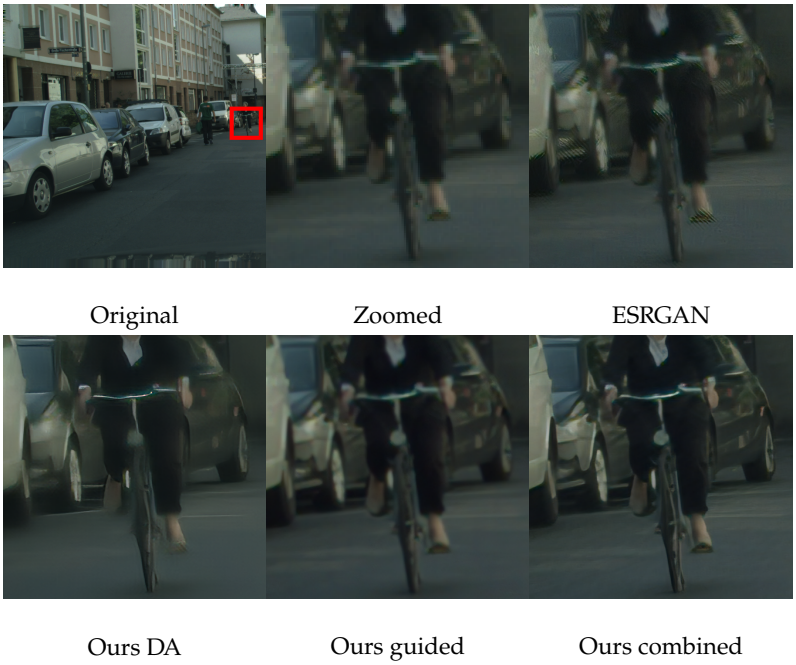


Fig. B.1: Super-resolution (×4) of a real image from the Cityscapes dataset [11]. By combining domain adaptation (DA) and guidance by semantic segmentation, our proposed method reconstructs visually pleasing images. In contrast, ESRGAN fails to handle the corruptions in the real image, resulting in many artifacts.

a SR model trained on such data to perform well on real LR images. Blind SR tries to address this problem by assuming an unknown downsampling kernel, but it still relies on Ground-Truth (GT) reference images for supervised learning.

Recent promising approaches to solve the Real-World Super-Resolution (RWSR) problem, where there aren't any LR/HR pairs for training, includes methods based on domain adaptation [17, 23, 43], where [17] was the winner of the NTIRE 2020 Challenge on RWSR [25]. These methods aim at creating synthetic LR images with similar characteristics as the real LR images. However, SR models relying solely on this approach are limited by the domain gap, due to the inherently challenging domain adaptation process. Self-learning based methods [3, 30] removes the need for paired training images, by learning an image specific SR model at test time, using only internal information available in the input image. However, this comes with a significant cost in terms of increased inference time [37].

In this work, we propose a novel framework, Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR), to handle SR of real LR images

without GT references or prior knowledge about the image formation model. We address the lack of training data by a combination of domain adaptation and guiding the SR learning by the loss of an auxiliary semantic segmentation network. Semantic Segmentation (SS) is a computer vision technique that provides scene understanding by dense labeling of pixels in an image. We argue that the loss of the SS task provides strong cues about the fidelity of the images, which can be used to jointly optimize the SR model towards producing more accurate, and noise-free HR images. The loss of the SS task also enables training on real LR images, without the need for GT reference image, which we argue can help the SR model adapt to the image-specific degradations. To reconstruct realistic textures, and ensure color consistency with the LR images, we propose to simultaneously train on synthetically generated LR/HR image pairs. To this end, we leverage domain adaptation to obtain LR images, with similar characteristics and corruptions as the real images. At test time, we decouple the SS network, which allows for faster inference times. To the best of our knowledge, we are the first to propose a framework for RWSR guided by the loss of a semantic segmentation network. We demonstrate the effectiveness of our proposed Semantic Segmentation Guided Real-World Super-Resolution (SSG-RWSR) on two publicly available datasets, using both real and synthesized LR images, and show that our method outperforms the existing SotA approaches. Visual results of our method can be seen in Figure B.1. In summary, the contributions of our work are as follows:

- We propose a novel framework for RWSR which allows learning from real LR images without requiring the corresponding GT images.
- We propose to guide the learning of the RWSR task with the loss of a semantic segmentation network, which helps to reconstruct sharp and noise-free HR images.
- We show that domain adaptation and guidance by the segmentation loss is complementary to each other, and improves the texture and fine details of the reconstructed images, compared to using guidance by the segmentation loss alone.
- Our method is trained end-to-end without any manual parameter tweaking.
- We show SotA results for RWSR on two publicly available datasets of both real and synthesized LR images.

2 Related work

2.1 Single image super-resolution

Current SotA methods for single image SR most often rely on deep Convolutional Neural Network (CNN) based SR architectures, which achieve impressive performance on artificially created LR images. Some of the most recent work includes EDSR [21], which is based on a deep residual CNN, the ResNet based SRResNet proposed by [20], and RCAN [42], which employs channel attention to re-scale features and recover HR details. These networks are optimized with MSE loss, which leads to good Peak Signal-to-Noise Ratio (PSNR) values, but fail to preserve the natural appearance of the images [41]. This problem is addressed in [20], which presents an SR model based on Generative Adversarial Networks (GANs), optimized with a combination of MSE, GAN, and VGG loss [18]. This approach leads to more photo-realistic images with better correlation to human perception of good image quality. In ESRGAN [36] this idea is further developed, mainly by improving the generator and adopting a relativistic discriminator. However, the performance of the aforementioned methods degrade significantly when used on real LR images [24]. This is mainly due to the domain gap between the real and synthetic LR images. To overcome this issue, ZSSR [29] introduced a zero-shot approach which learns an image specific SR model at test time. In MSZR [30] this concept is extended to exploit information from an external dataset as well. In KernelGAN [3], ZSSR is used together with a GAN based network for estimation of image-specific blur kernels. DAN [26] proposed to address both steps in a single model using an alternating optimization algorithm that jointly estimates blur kernels and performs SR. However, these image-specific learning methods come with the cost of extremely slow prediction times compared to other SR methods [37]. In contrast, the prediction times of our method are similar to [36]. In [17], a domain adaptation based approach to RWSR is presented. First, a pool of realistic blur-kernels and noise patches is collected. These are then used to transform clean HR images into realistic LR images with similar appearance as real LR images. Next, a SR model is trained on the constructed data. However, since the domain adaptation is a challenging task in itself, the SR model is limited by the domain gap between the synthesized and real LR images. In DPSR [39], de-blurring and de-noising are combined with SR to deal with blurry and noisy LR images. However, without sufficient prior information about the image-specific degradations, the effectiveness of the method is limited.

2.2 Guided super-resolution

Lutio et al. [12], proposed a method for super-resolution of depth images guided by RGB images. By considering it a pixel-to-pixel transformation problem, they learn a mapping between the LR and HR images that are also applicable to the depth image. Inversely [10] proposed a zero-shot approach that extracts LR and HR patches using corresponding depth maps. Subsequently they train a GAN that employs SR- and Degredation Simulation Network (DSN)-modules in a cyclical manner that alternates between $LR \rightarrow HR \rightarrow LR$ and $HR \rightarrow LR \rightarrow HR$ mapping. In image generation tasks, such as [9, 16, 45] it has been shown that semantic information can be utilized to generate detailed textures and realistic looking images. In [27], semantic information is used to guide a SR network towards creating textures in areas where this is important, and creating sharper lines at object boundaries. Condition networks that employ SS probability maps to actively guide the SR network at a feature-map level is proposed in [35] and [22]. It is shown in [35] that the conditions can strongly influence the textures generated and result in much more realistic looking textures that are more semantically appropriate. While [27] shows that CNNs learn some categorical information, [28] propose that more categorical information can be learned by treating SR as a multi-task problem where a parallel network head that predicts a semantic map is added. The shared backbone is then forced to learn the categorical information necessary for accurate segmentation, which benefits the SR head. The work most closely related to ours is [40], which use multi-task learning to jointly perform SS and SR, and control the balance between SS and SR performance by adaptive weighting. However, when the SR task is given the highest weight, the performance does not benefit much from the semantic information, and drops further as more priority is given to the SS task. Furthermore, a key difference from this, and all of the existing methods utilizing semantic information for SR, is that they require paired LR and HR images for training, which makes them unsuitable for the RWSR problem. On the contrary, we show that semantic information can be leveraged to solve the RWSR problem where no GT reference images are available, making our method applicable to scenarios where real-world images, such as the ones from surveillance cameras, need to be improved by super-resolution.

2.3 Semantic segmentation

Much like in SR, SS architectures tends to follow an encoder-decoder architecture, that first encodes information with feature extraction network, typically a ResNet variant, and then decodes it again to recover spatial information and resolution. Learning to recover spatial information is difficult [7, 38], and as such SotA SS methods have tended towards architectures that retain spatial

resolution to some extent. PSPNet [38] proposed using a pyramid pooling module where the input feature-map would be pooled across different regions varying from 1×1 to 6×6 sub-regions, to get varying degrees of detail in the pooled feature-maps. They further employ 1×1 convolution to reduce the channel depth before concatenation. To recover the initial resolution lost from repeated convolution, the feature-maps are upsampled with bilinear interpolation to match the original input size. DeepLabv3 [7] proposed using atrous-convolution in the encoder to create coarse feature-representations before employing a spatial pooling pyramid to recover information at different scales. This was further expanded in [8] with depth-wise-separable convolutions resulting in the network being able to learn more fine-grained control of the details in each layer. HRNet [33] proposed an architecture that retains the spatial resolution of one branch, and parallel branches that perform further convolutions, rather than sequential repeated convolutions. Retaining the resolution with further convolutions in a parallel branch allows for the retention of fine-grained detail, while still obtaining deep representational information. However while HRNet attempts to keep a higher resolution, the initial convolutions result in an output prediction which is one-fourth of the size of the input image, which means that the prediction has to be up-sampled to compute the prediction accuracy. By super-resolving the input image, the need for up-sampling of the prediction is avoided, which leads to more accurate predictions [1], which in turn improves the guiding of a SR network by the semantic loss. In [34], an auxiliary super-resolution branch is used to improve the performance on a semantic segmentation model. The SS model shares encoder weights with the SR model, which are optimized during training with MSE loss, before being removed at test time. The training process requires paired LR and HR images, and the method is therefore not applicable to real-world applications.

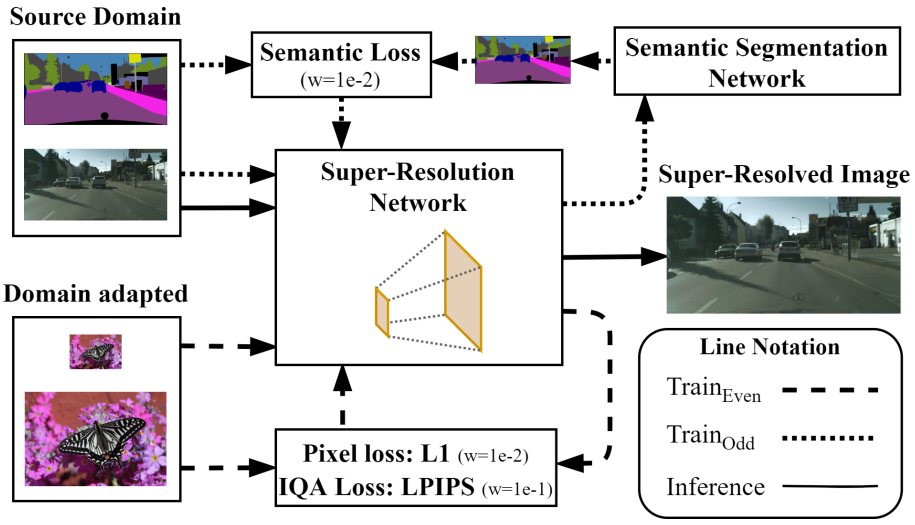


Fig. B.2: Schematic overview of our proposed SSG-RWSR. To learn to perform RWSR we leverage both guiding from an auxiliary semantic segmentation task and domain adaptation. At test time, the semantic segmentation network is de-coupled, and as such no semantic labels are required to super-resolve the LR test images.

3 The proposed method

The fundamental challenge in RWSR is the lack of real natural LR/HR image pairs which can be used to learn a SR network with supervised learning. Current RWSR methods often constrain the SR problem by assuming that the LR image is the result of an imaging model described as:

$$I_{LR} = (I_{HR} * k) \downarrow_s + n \tag{B.1}$$

where k, s, and n denotes blur kernel, scaling factor, and noise, respectively. However, in reality, the image formation of real images is much more complicated.

A block diagram of our proposed SSG-RWSR framework can be seen in Figure B.2. We propose to combine domain adaptation and guiding of the SR learning by the loss of an auxiliary semantic segmentation network. The benefit of guiding the SR learning by the segmentation loss is two-fold. First, this helps our SR network to adapt to the natural image characteristics of the LR images in the source domain, without the need for GT reference images. This is important as these can be cumbersome, and sometimes even impossible to obtain. Conversely, LR images can always be annotated with semantic labels. Secondly, the loss of the segmentation task can provide strong cues about the level of noise in the images, and the quality of object boundaries that can help guide the SR network towards producing more accurate reconstructions.

We support the SR learning by training on image pairs created with domain adaptation. This helps our model to reconstruct realistic textures and accurate colors. During training, we alternate between training on real LR images in the source domain X, guided by SS, and LR images created by our domain adaptation approach, to leverage information from both domains. Both concepts are elaborated in the following subsections.

3.1 Guiding with semantic segmentation

We argue that a SS model can benefit from input images with low noise and high levels of detail, which can be provided by a carefully trained SR model. Hence the accuracy of a SS model can be used to guide the SR network towards producing better image quality. Based on this assumption, we structure our SSG-RWSR such that the SS network is fully dependant on the SR output. This is different from [28], where a separate semantic head is used, as we argue that for optimal guidance, the two networks should be directly linked. During training on real images, the input LR image is sequentially processed by the SR and SS networks. The SS loss is then used to optimize both the SR and SS models. This means that the SR model is getting increasingly better at producing HR images that are optimal for the segmentation task, and in addition, the SS model continuously adapts to the improved input images to further optimize the segmentation accuracy.

3.2 Domain adaptation

To ensure that our SR network learns to reconstruct HR images with realistic textures and maintain consistency with the LR input images in terms of color, we also train our SR model on paired LR/HR images. To obtain LR images with similar image characteristic as the real LR images in the source domain X, we utilize domain adaptation [15]. The procedure is elaborated in the following.

Estimation of degradation parameters We map clean HR images from the target domain Y to the real LR source domain X to minimize the domain gap between real and synthesized LR images. Our approach is based on kernel estimation and sampling of realistic noise patches [17]. For estimation of realistic blur kernels, we use KernelGAN [3], on real LR images in X to build a pool of image-specific blur kernels that can be used to degrade the clean HR images in Y.

To generate artificial LR images which are more similar to the real LR images we employ the method from [6] to sample noise from the real LR images in X. This approach assumes that realistic noise can be obtained from an image by extracting patches from uniform areas, and then subtracting the mean. To this end, we define two patches p_i and q_j^i. p_i is obtained by a sliding

window approach across images in X. Similarly q_j^i is obtained by scanning p_i. We consider p_i a uniform patch if the following constraints are met:

$$|Mean(q_j^i) - Mean(p_i)| \leq \mu \cdot Mean(p_i) \tag{B.2}$$

and

$$|Var(q_j^i) - Var(p_i)| \leq \gamma \cdot Var(p_i) \tag{B.3}$$

where $Mean$ and Var denote the mean and variance, respectively, and μ and γ are scaling factors. Different from [6] we add an additional constraint to ensure that saturated patches are not extracted:

$$Var(p_i) \geq \phi \tag{B.4}$$

where ϕ denotes a minimum variance threshold. If all constraints are satisfied p_i is considered a valid noise patch, from which we subtract the mean value and then add to a pool of noise patches n_i.

Realistic image degradation We degrade clean HR images from the target domain Y with the estimated blur kernels and noise patches following the image formation model described in Equation 3. More specifically, we create artificial LR images I_D, by first convolving a HR image in Y with a randomly selected kernel k_i from the pool of estimated blur kernels, followed by a downsampling operation. The process can formally be described as:

$$I_D = (Y_n * k_i) \downarrow_s, i \in \{1, 2 \cdots m\} \tag{B.5}$$

where I_D is the downscaled image, Y_n is a HR image, k_i refers to a kernel from the degradation pool $\{k_1, k_2, \cdots k_m\}$ and s is the scaling factor.

During training of our SR network, we inject noise to the synthesized LR images by applying a randomly selected noise patch from the pool of noise patches n_i. The processes can be described as:

$$I_N = I_D + n_i, i \in \{1, 2 \cdots l\} \tag{B.6}$$

where I_D is a downscaled image, and n_i is a noise patch from the noise pool $\{n_1, n_2, \cdots n_l\}$.

3.3 Backbone networks

Super-resolution Our SR network consist of 23 Residual-in-Residual Dense Blocks (RRDBs) [36]. To better utilize the semantic information we use a LR patch size of 128×128 pixels. We use a combination of L1 pixel loss, $\mathcal{L}_{pix}$, and Learned Perceptual Image Patch Similarity (LPIPS) loss, $\mathcal{L}_{lpips}$, to optimize the network when training on the domain adapted images. The L1 loss ensures

color consistency between the prediction and the GT image, while LPIPS loss helps to improve the perceptual quality with strong correlation to human perception [41]. The total loss for learning the SR model from the domain adapted images is defined as:

$$\mathcal{L}_{domain-adapted} = \lambda_{pix} \cdot \mathcal{L}_{pix} + \lambda_{lpips} \cdot \mathcal{L}_{lpips} \tag{B.7}$$

where λ_{pix}, and λ_{lpips} are scaling parameters.

Semantic segmentation To maintain a high spatial resolution throughout the segmentation network we use an architecture with multiple parallel high-to-low resolution subnetworks with information exchange [33] as our SS backbone. We optimize the segmentation model with cross-entropy loss, $\mathcal{L}_{ce}$, which is also used for guiding the SR model. The loss for guiding the SR learning is defined as:

$$\mathcal{L}_{guided} = \lambda_{ce} \cdot \mathcal{L}_{ce} \tag{B.8}$$

where λ_{ce} is a scaling parameter.

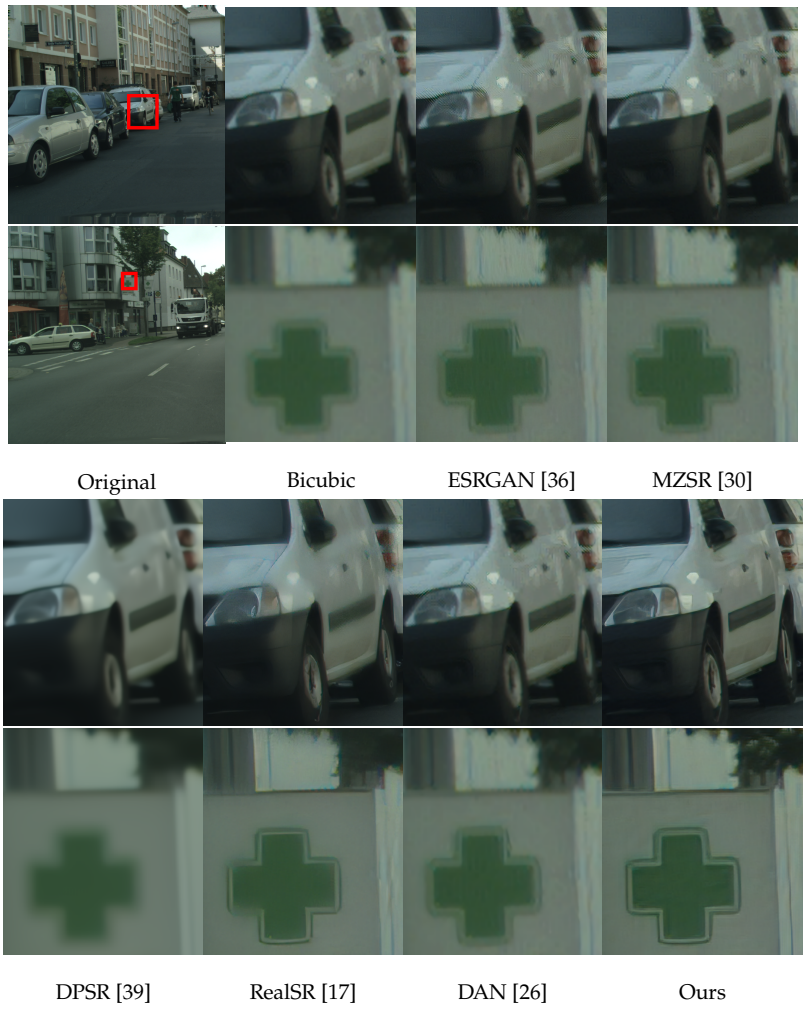


Fig. B.3: Comparison with SotA methods for ×4 SR of **real** images from the Cityscapes dataset. As visible, our method reconstructs sharper and more visually appealing results compared to the existing methods.

Fig. B.4: Comparison with SotA methods for ×4 of **synthetically** degraded images from the Cityscapes dataset. As visible, our method reconstructs sharp images with low noise compared to the existing methods.

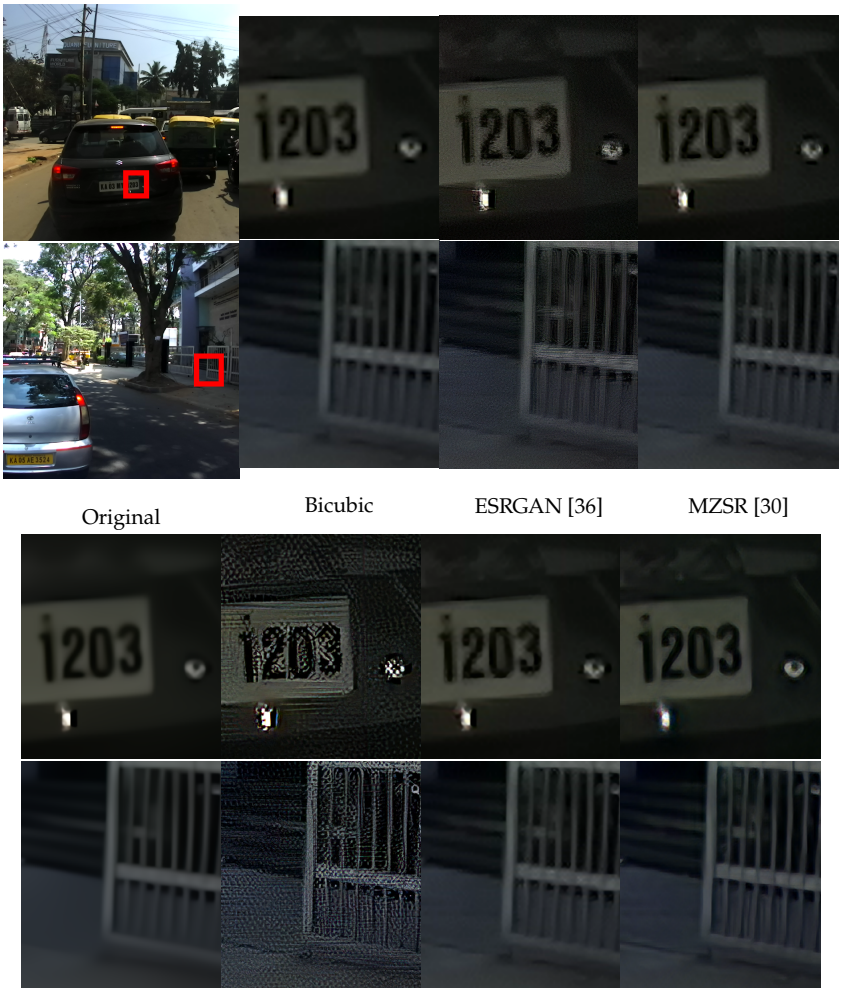


Fig. B.5: Comparison with SotA methods for ×4 SR of **real** images from the IDD dataset. As visible, our method reconstructs more detailed images with less artifacts compared to the existing methods.

4 Implementation details

Similar to recent RWSR literature [5, 24, 25] we perform our experiments with $\times 4$ scaling factor. For the creation of realistic training image pairs, as described in Section 3.2, we use the DF2K dataset as target domain Y of clean HR images. The DF2K is a merge of 800 and 2650 images from DIV2K [2] and Flickr2K [31], respectively.

Training details To train our SR and SS backbones, we initialize from models pre-trained on DF2K and Cityscapes, respectively. We jointly train both models, alternating between updating both models based on the cross-entropy loss, and updating only the SR model based on pixel and LPIPS loss. We denote the two update cycles as $Train_{Odd}$ and $Train_{Even}$ respectively. We use a batch size of 12 and train for 100000 iterations on randomly cropped LR patches and semantic labels using four V100 GPUs. We use the ADAM optimizer with an initial learning rate of 1×10^{-4} for both models. Through experimentation, we find suitable weights for the loss functions and set λ_{pix}, λ_{lpips}, λ_{ce} to 0.01, 0.1, and 0.01 respectively. For extraction of realistic noise patches from X, we set p_i to match the LR patch size and set q_j^i to 32, μ to 0.1, γ to 0.3, and ϕ to 0.5 which we find appropriate for real images.

Inference At test time, we de-couple the segmentation network, and as such, semantic labels are no longer required. We obtain super-resolved images by running our trained SR on the full LR input image. Hence the inference time of our SSG-RWSR is similar to [36].

5 Experiments and results

We compare our proposed method to four recent SotA methods for SR of real images, namely MZSR [30], DPSR [39], RealSR [17], and DAN [26]. We adjust the competing models for optimal performance for a fair comparison. We use KernelGAN [3] to estimate blur kernels for use with MZSR [30]. For DPSR [39] and DAN [26], we set noise levels as recommended by the authors. With RealSR [17] we use the degradation framework provided by the authors, and re-train the model to the respective datasets. We also include the ESRGAN [36] in our comparison, to highlight the effect of applying a SR model trained on bicubically downsampled LR images on real LR images. For this, we use the pre-trained weights provided by the authors.

5.1 Datasets

Evaluation on real images For evaluation on real images we use the Cityscapes [11] and IDD [32] datasets, which both contain images and appertaining semantic labels. The Cityscapes dataset has 19 different classes and is divided

into 2975 training, 500 validation, and 1525 test images, respectively, which have a resolution of 2048×1024 pixels. We use the validation set to evaluate the performance of our method. The IDD dataset has 30 different classes and contains both images of 1920×1080 and 1280×720 pixels. For our experiments, we use the 1280×720 pixels images from the training and validation set which amount to 1876 and 442 images respectively.

Evaluation on synthesized images To validate the performance of the proposed SSG-RWSR on images with known GTs, we conduct experiments on synthetically degraded LR images. This allows for evaluation with Full-Reference Image Quality Assessment (FR-IQA) metrics. To simulate realistic LR images we first degrade the images by convolving an 11×11 Gaussian blur kernel with a standard deviation of 1.5 before downsampling. Following the protocol from [24], we model sensor noise by adding Gaussian noise, with zero mean and a standard deviation of 8 pixels. This simulates real-world LR images acquired with a low-quality camera, in poor lighting conditions. For consistency, we also downsample the appertaining semantic labels. During training, only the degraded LR images and labels are available, and the degradation process and GTs are kept hidden. We perform our experiments with synthesized LR images on the Cityscapes dataset.

5.2 Quantitative Evaluation metrics

Due to the lack of GT reference images, it impossible to compare the reconstruction performance on real images with traditional SR FR-IQA metrics. As such we mainly rely on Mean Opinion Rank (MOR), which is a direct measure of human perceived perceptual quality [25]. We ask the participants to rank the super-resolved images based on overall image quality. We randomly shuffle the presented images to avoid bias. Readers can refer to our supplementary material for more details about our evaluation with MOR. Furthermore, we also evaluate the performance using two SotA learning based No-Reference Image Quality Assessment (NR-IQA) methods, namely, NIMA [14] and MetaIQA [44] as these show a good correlation to human judgement. For both methods, we use the pre-trained weights for evaluation of the technical image quality.

For our experiments on synthesized LR images, we use two traditional SR metrics, PSNR and SSIM, and two perceptually oriented metrics, LPIPS [41], and DISTS [13]. Out of these, we mainly consider the LPIPS and DISTS metrics as indicators of the image quality due to their high correlation with human judgement [41]. Note that low distortion and high perceptual quality are at odds with each other, making it impossible to two obtain both [4]. With the use of GAN training and perceptual loss, our method is optimized to obtain a good trade-off with a slight bias towards perceptual quality.

Cityscapes (Real LR images)			
Method	NIMA ↑	Meta-IQA ↑	MOR ↓
Bicubic [19]	4.62	0.245	-
ESRGAN [36]	4.95	0.247	-
MZSR [30]	4.88	0.231	3.33
DPSR [39]	4.83	0.240	4.41
RealSR [17]	4.87	0.236	2.75
DAN [26]	4.65	0.246	3.47
Ours	**5.04**	**0.254**	**1.21**

Table B.1: Quantitative results on the Cityscapes validation sets. ↑ and ↓ indicate whether higher or lower values are desired, respectively. As seen, our method obtains both the best MOR and NIMA and Meta-IQA results.

IDD (Real LR images)			
Method	NIMA ↑	Meta-IQA ↑	MOR ↓
Bicubic [19]	4.73	0.330	-
ESRGAN [36]	4.94	0.325	-
MZSR [30]	5.00	0.330	2.96
DPSR [39]	4.92	0.330	3.16
RealSR [17]	4.83	0.296	4.88
DAN [26]	4.77	**0.330**	2.48
Ours	**5.03**	0.323	**1.45**

Table B.2: Quantitative results on the IDD validation sets. ↑ and ↓ indicate whether higher or lower values are desired, respectively. As seen, our method obtains both the best MOR and NIMA results, and the second best Meta-IQA results.

5.3 Quantitative results

Real images As show in Table B.1 and B.2 our method results in the most visually pleasing reconstructions of both real images from the CityScapes and IDD datasets according to the MOR. This is also supported by the NIMA and Meta-IQA scores, where only the DAN [26] is slightly better according to the Meta-IQA scores on the IDD dataset. However, this is in contrast to the visual appearance of the images, as the digits on the licence plates shown in Figure B.5 are more well defined in the image produced by our method, compared to the ones produced by DAN.

Synthesized images As shown in Table B.3 our method achieves a good compromise between fidelity and perceptual quality, by obtaining the best LPIPS and DISTS scores, which indicate that our super-resolved images are closer to the GT in terms of visual quality, and the second best results on the

Cityscapes (Synthesized LR images)				
Method	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Bicubic [19]	27.51	0.62	0.64	0.19
ESRGAN [36]	18.17	0.12	1.29	0.20
MZSR [30]	26.68	0.55	0.73	0.16
DPSR [39]	**33.11**	**0.90**	0.42	0.13
RealSR [17]	25.88	0.77	0.26	0.10
DAN [26]	27.16	0.58	0.60	0.20
Ours	29.08	0.83	**0.19**	**0.07**

Table B.3: Quantitative results on the artificially degraded Cityscapes validation set. ↑ and ↓ indicate whether higher or lower values are desired, respectively. Our method achieves a good trade-off between low distortion and high perceptual quality with the second best PSNR and SSIM results, and the best perceptual quality as measured by the LPIPS and DISTS metrics.

hand-crafted metrics (PSNR, SSIM). The latter is expected, as our method is optimized towards perceptual image quality, which are at odds with a low reconstruction error [4].

5.4 Qualitative results

Real images In Figure B.3 and B.5 we visualize super-resolution results of real LR images. We see that most methods fail to handle the highly complex degradation process present in the real images, which results in many artifacts (ESRGAN, MZSR, RealSR) or blurry images (DPSR, DAN). In comparison, our method generates sharper images with better visual quality and less noise.
Synthesized images In Figure B.4 we see that ESRGAN, MZSR and DAN cannot properly handle the noisy LR image which causes a high degree of artifacts to be present in the super-resolved images. DPSR performs better in that regard, but the images appear blurry and lack high-frequency details. In contrast, both RealSR and our method produces artifact-free, sharp, and natural appearing images.

5.5 Ablation study

To study the effect of the individual components in our proposed SSG-RWSR framework we compare ablations of the framework to the full system. Figure B.1 and Table B.4 shows the visual difference, and quantitative results for the different settings, respectively. As seen, training only on the synthetically created LR/HR pairs results in HR images with more high-frequency details than the LR image. However in some areas, the hallucinated details appear to be incorrect or missing. On the contrary, training only on the real LR images

guided by the SS loss, produces less detailed images, but the reconstructions are more consistent with the objects and shapes present in the LR image. In comparison, our combined SSG-RWSR produces images that are both sharp, detail rich, and with a photo-realistic appearance.

Method	NIMA ↑	Meta-IQA ↑
Ours (DA)	4.33	0.206
Ours (Guided only)	5.00	0.251
Ours	**5.04**	**0.254**

Table B.4: The effect of the different components in our proposed method on the Cityscapes validation set. ↑ and ↓ indicate whether higher or lower values are desired, respectively.

6 Conclusion

In this paper, we address the RWSR problem where no ground truth data are available. To this end, we introduce a novel framework, SSG-RWSR, where the SR learning is guided by an auxiliary semantic segmentation network. This enables our SR model to adapt to the image specific degradations present in real LR images, and enables reconstruction of sharp object boundaries and noise-free images. We combine guidance by the segmentation loss with domain adaptation, to reconstruct realistic textures and ensure color consistency. Our experimental results on both real and synthesized LR images demonstrate a significant improvement over the SotA methods, resulting in less noise and better visual quality. This is supported by human ranking of the super-resolved images, where our method outperforms other methods by large margins.

Disclosure of Funding This research was funded by Milestone Systems A/S, Brøndby Denmark and the Independent Research Fund Denmark, under grant number 8022-00360B.

References

[1] A. Aakerberg, A. Johansen, K. Nasrollahi, and T. Moeslund, "Single-loss multi-task learning for improving semantic segmentation using super-resolution," *In Press, CAIP*, 2021.

[2] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *CVPR*, 2017.

[3] S. Bell-Kligler, A. Shocher, and M. Irani, "Blind super-resolution kernel estimation using an internal-gan," in *NIPS*, 2019.

[4] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *CVPR*. IEEE, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.html

[5] J. Cai, S. Gu, R. Timofte, and L. Zhang, "Ntire 2019 challenge on real image super-resolution: Methods and results," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[6] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *CVPR*, 2018.

[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017.

[8] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[9] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *ICCV*, 2017.

[10] X. Cheng, Z. Fu, and J. Yang, "Zero-shot image super-resolution with depth guided internal degradation learning," in *ECCV*, 2020.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *CVPR*, 2016.

[12] R. De Lutio, S. D'Aronco, J. D. Wegner, and K. Schindler, "Guided super-resolution as pixel-to-pixel transformation," in *ICCV*, 2019.

[13] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *CoRR*, 2020. [Online]. Available: https://arxiv.org/abs/2004.07728

[14] H. T. Esfandarani and P. Milanfar, "NIMA: neural image assessment," *TIP*, 2018.

[15] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *ICCV*, 2013.

[16] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, 2017.

[17] X. Ji, Y. Cao, Y. Tai, C. Wang, J. Li, and F. Huang, "Real-world super-resolution via kernel estimation and noise injection," in *CVPR Workshops*, 2020.

[18] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *ECCV*, 2016.

[19] R. G. Keys, "Cubic Convolution Interpolation for Digital Image Processing," *IEEE Trans Acoust. Speech Signal Process*, 1981.

[20] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, 2017.

[21] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPR Workshops*, 2017.

[22] L. Liu, S. Wang, and L. Wan, "Component semantic prior guided generative adversarial network for face super-resolution," *IEEE Access*, 2019.

[23] A. Lugmayr, M. Danelljan, and R. Timofte, "Unsupervised learning for real-world super-resolution," in *CVPR Workshops*.

[24] A. Lugmayr, M. Danelljan, R. Timofte, M. Fritsche, S. Gu, K. Purohit, P. Kandula, M. Suin, A. N. Rajagoapalan, N. H. Joon, Y. S. Won, G. Kim, D. Kwon, C. Hsu, C. Lin, Y. Huang, X. Sun, W. Lu, J. Li, X. Gao, S. Bell-Kligler, A. Shocher, and M. Irani, "Aim 2019 challenge on real-world image super-resolution: Methods and results," in *ICCV Workshops*, 2019.

[25] R. Lugmayr, M. Danelljan, and R. Timofte, "Ntire 2020 challenge on real-world image super-resolution: Methods and results," *CVPR Workshops*, 2020.

[26] Z. Luo, Y. Huang, S. Li, L. Wang, and T. Tan, "Unfolding the alternating optimization for blind super resolution," in *NeurIPS*, 2020.

[27] M. S. Rad, B. Bozorgtabar, U. Marti, M. Basler, H. K. Ekenel, and J. Thiran, "SROBB: targeted perceptual loss for single image super-resolution," in *ICCV*, 2019.

[28] M. S. Rad, B. Bozorgtabar, C. Musat, U. Marti, M. Basler, H. K. Ekenel, and J. Thiran, "Benefiting from multitask learning to improve single image super-resolution," *Neurocomputing*, 2020.

[29] A. Shocher, N. Cohen, and M. Irani, ""zero-shot" super-resolution using deep internal learning," in *CVPR*, June 2018.

[30] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *CVPR*, 2020.

[31] R. Timofte, E. Agustsson, L. Van Gool, M. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *CVPR Workshops*, 2017.

[32] G. Varma, A. Subramanian, A. M. Namboodiri, M. Chandraker, and C. V. Jawahar, "IDD: A dataset for exploring problems of autonomous navigation in unconstrained environments," in *WACV*, 2019.

[33] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2019.

[34] L. Wang, D. Li, Y. Zhu, L. Tian, and Y. Shan, "Dual super-resolution learning for semantic segmentation," in *CVPR*, 2020, pp. 3774–3783.

[35] X. Wang, K. Yu, C. Dong, and C. C. Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *CVPR*, 2018.

[36] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *ECCV*, 2019.

[37] Z. Wang, J. Chen, and S. Hoi, "Deep learning for image super-resolution: A survey," *TPAMI*, 2020.

[38] H. Z., J. S., X. Q., X. W., and J. J., "Pyramid scene parsing network," in *CVPR*, 2017.

[39] K. Zhang, W. Zuo, and L. Zhang, "Deep plug-and-play super-resolution for arbitrary blur kernels," in *CVPR*, 2019.

[40] Q. Zhang, G. Yang, and G. Zhang, "Collaborative network for super-resolution and semantic segmentation of remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–12, 2021.

[41] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html

[42] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., 2018.

[43] R. Zhou and S. Süsstrunk, "Kernel modeling super-resolution on real low-resolution images," in *ICCV*, 2019.

[44] H. Zhu, L. Li, J. Wu, W. Dong, and G. Shi, "Metaiqa: Deep meta-learning for no-reference image quality assessment," in *CVPR*, 2020.

[45] S. Zhu, R. Urtasun, S. Fidler, D. Lin, and C. Change Loy, "Be your own prada: Fashion synthesis with structural coherence," in *ICCV*, 2017.

Paper C

Video Transformers: A Survey

Javier Selva, Anders S. Johansen, Sergio Escalera, Kamal Nasrollahi, Thomas B. Moeslund and Albert Clapés

The paper has been published in the
IEEE Transactions on Pattern Analysis and Machine Intelligence., pp. 1–20, 2023.

The layout has been revised.

Abstract

Transformer models have shown great success handling long-range interactions, making them a promising tool for modeling video. However, they lack inductive biases and scale quadratically with input length. These limitations are further exacerbated when dealing with the high dimensionality introduced by the temporal dimension. While there are surveys analyzing the advances of Transformers for vision, none focus on an in-depth analysis of video-specific designs. In this survey, we analyze the main contributions and trends of works leveraging Transformers to model video. Specifically, we delve into how videos are handled at the input level first. Then, we study the architectural changes made to deal with video more efficiently, reduce redundancy, re-introduce useful inductive biases, and capture long-term temporal dynamics. In addition, we provide an overview of different training regimes and explore effective self-supervised learning strategies for video. Finally, we conduct a performance comparison on the most common benchmark for Video Transformers (i.e., action classification), finding them to outperform 3D ConvNets even with less computational complexity.

1 Introduction

Video is increasingly becoming a popular medium to convey audio-visual information. Video provides the visual appeal of images while introducing motion and deformations through the additional time dimension. As such, processing video data is partially akin to both images (continuous visual signals) and natural language processing (structured as a sequence). The video domain further introduces its own challenges, namely a large increase in dimensionality linked with a high level of information redundancy and the need to model motion dynamics.

Transformers [183] are a recent family of models, originally designed to replace recurrent layers in a machine translation setting. Its purpose was to remedy limitations of sequence modeling architectures by handling whole sequences at once (as opposed to RNNs, which are sequential in nature), allowing further parallelization. Besides, it removes the locality bias of traditional architectures, such as CNNs, and instead learns interactions of non-local contexts of the input. This lack of inductive biases makes Transformers very versatile, as seen by the quick adoption for modeling many data types [12, 32, 34, 52, 132], including videos [2, 5, 49, 94, 109, 239]. The Transformer evolves input representations based on interactions among all sequence elements. These interactions are modulated through a pair-wise affinity function that weighs the contribution that every element should have on any other. The ability to model all-to-all relationships can be especially beneficial to understand motion cues,

long-range temporal interactions, and dynamic appearance changes in video data. However, Transformers scale quadratically with sequence length T (i.e., $O(T^2)$, due to the pair-wise affinity computation) which is exacerbated by the high dimensionality of video. Furthermore, the lack of inductive biases makes Transformers require large amounts of data or several modifications to adapt to the highly redundant spatiotemporal structure of video.

The recent surge in *Video Transformer* (VT) works makes it convoluted to keep track of the latest advances and trends. Existing surveys focus on design choices for Transformers in general [102], NLP [82], images [105, 209], or efficient designs [44, 175]. Given the sequential nature of video, as well as the large dimensionality and redundancy introduced by the temporal dimension, adopting image-based solutions or NLP-based designs for long-term modeling will not suffice. While other existing surveys include video, they are limited to superficial comments of a few VTs in the broader context of vision Transformers [59, 83, 214], techniques to integrate visual data with other modalities [164, 207], or video-language pre-training [153]. In this sense, they miss an in-depth analysis that properly captures the challenges of modeling raw image sequences or highly redundant spatiotemporal visual features through Transformers.

In this survey, we comprehensively analyze advances and limitations of Transformers when considering the particularities of modeling video data. To do so, we review over 100 VT works and provide detailed taxonomies of the various design choices throughout the VT pipeline (namely input, architecture, and training). Finally, we extensively compare performance on the task of video classification based on self-reported results from the state-of-the-art on Kinetics 400 [15] and Something-Something-v2 [113].

The structure of the paper is as follows: appendix 2 introduces the original Transformer; in appendix 3 we explore how videos are handled prior to the Transformer; appendix 4 describes Transformer design adaptations to video; appendix 5 investigates common training strategies; appendix 6 discusses VTs performance on action classification; and in appendix 7 we discuss the main trends, limitations, and future work. For an extensive list of all VT works reviewed in this survey, and details on how each section in this survey relates to a given work, see appendix 8 in the supplementary.

2 The Transformer

Originally proposed for language translation [183], the Transformer consists of two distinct modules: encoder and decoder, each composed of several stacked Transformer layers (see fig. C.1). The *encoder* was designed to produce a representation of the source language sentence that is then attended by the *decoder*, which will eventually translate it into the target language. We first

introduce a few necessary concepts (input pre-processing and the self-attention operation) to then follow the flow of the Transformer while explaining its components and functioning.

Input pre-processing: tokenization, linear embedding, and positional encodings. The *tokenization* converts the input source and target language sentences into sequences of words (or subwords), namely "tokens". Let $\widetilde{\widetilde{\mathbf{X}}} = (\widetilde{\widetilde{\mathbf{x}}}_1, \dots, \widetilde{\widetilde{\mathbf{x}}}_{N_\mathrm{x}})$ and $\widetilde{\widetilde{\mathbf{Z}}} = (\widetilde{\widetilde{\mathbf{z}}}_1, \dots, \widetilde{\widetilde{\mathbf{z}}}_{N_\mathrm{z}})$ be, respectively, the source and target sequences of one-hot encoded tokens over their respective word vocabularies $\mathcal{X}$ and $\mathcal{Z}$ (i.e., $\widetilde{\widetilde{\mathbf{x}}} \in \mathbb{R}^{|\mathcal{X}|}$ and $\widetilde{\widetilde{\mathbf{z}}} \in \mathbb{R}^{|\mathcal{Z}|}$). Then, *linear embedding* is simply the step of projecting one-hots to a continuous embedding space via a learned linear transformation: $f_\mathcal{X} : \mathbb{R}^{|\mathcal{X}|} \mapsto \mathbb{R}^{d_m}$ (analogously $f_\mathcal{Z}$), where d_m will be the dimensionality handled internally by the Transformer. This way, we obtain the source embeddings $\widetilde{\mathbf{X}} = (f_\mathcal{X}(\widetilde{\widetilde{\mathbf{x}}}_1), \dots, f_\mathcal{X}(\widetilde{\widetilde{\mathbf{x}}}_{N_\mathrm{x}}))$ and target embeddings $\widetilde{\mathbf{Z}} = (f_\mathcal{Z}(\widetilde{\widetilde{\mathbf{z}}}_1), \dots, f_\mathcal{Z}(\widetilde{\widetilde{\mathbf{z}}}_{N_\mathrm{z}}))$. Finally, *positional encodings* are added to signal the position of the tokens in the sequence to the later (otherwise permutation invariant) attention operations. Defined using a set of (non-learnable) sinusoidal encodings (see [183] for details), these are added to the source/target embeddings before being input to encoder/decoder (as depicted in fig. C.1): $\mathbf{X}^0 = (\widetilde{\mathbf{x}}_1 + \mathbf{e}^\mathrm{x}_1, \dots, \widetilde{\mathbf{x}}_{N_\mathrm{x}} + \mathbf{e}^\mathrm{x}_{N_\mathrm{x}})$ and $\mathbf{Z}^0 = (\widetilde{\mathbf{z}}_1 + \mathbf{e}^\mathrm{z}_1, \dots, \widetilde{\mathbf{z}}_{N_\mathrm{z}} + \mathbf{e}^\mathrm{z}_{N_\mathrm{z}})$, where $\mathbf{e}^\mathrm{x}_\cdot, \mathbf{e}^\mathrm{z}_\cdot \in \mathbb{R}^{d_\mathrm{m}}$.

Self-attention (SA). It is the core operation of the Transformer. Given an arbitrary sequence of token embeddings $\mathbf{X} \in \mathbb{R}^{N_\mathrm{x} \times d_\mathrm{m}}$ (e.g., $\mathbf{X}^0$), it augments (contextualizes) each of the embeddings $\mathbf{x}_i \in \mathbb{R}^{d_\mathrm{m}}$ with information from the rest of embeddings. For that, the embeddings in $\mathbf{X}$ are linearly mapped to the embedding spaces of *queries* $\mathbf{Q} = \mathbf{X}\mathbf{W}_\mathrm{Q} \in \mathbb{R}^{N_\mathrm{x} \times d_\mathrm{k}}$, *keys* $\mathbf{K} = \mathbf{X}\mathbf{W}_\mathrm{K} \in \mathbb{R}^{N_\mathrm{x} \times d_\mathrm{k}}$, and *values* $\mathbf{V} = \mathbf{X}\mathbf{W}_\mathrm{V} \in \mathbb{R}^{N_\mathrm{x} \times d_\mathrm{k}}$, where $\mathbf{W}_\mathrm{Q}, \mathbf{W}_\mathrm{K} \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_\mathrm{V} \in \mathbb{R}^{d_m \times d_v}$, and typically $d_\mathrm{k}, d_\mathrm{v} <= d_m$. Then, self-attention can be computed as follows:

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_\mathrm{k}}}\right)\mathbf{V}. \tag{C.1}$$

The dot-product $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N_\mathrm{x} \times N_\mathrm{x}}$ is a measure of similarity. Intuitively, the larger the similarity between $\mathbf{q}_i \in \mathbf{Q}$ and $\mathbf{k}_j \in \mathbf{K}$ the more relevant the information embedded in $\mathbf{x}_j$ is for $\mathbf{x}_i$. However, this aggregation is not done in the space of $\mathbf{X}$, but in the one of the values. By applying Softmax with temperature $\sqrt{d_k}$, we come up with normalized similarities (the self-attention matrix) that weigh how much each of the values $\mathbf{v}_j$ contributes to the output representation of every other $\mathbf{v}_i$.

Encoder module. It consists of E layers, each including *Multi-Head Self-Attention* (MHSA) and *Position-wise Feed-Forward Network* (PFFN) sub-layers. The MHSA sub-layer performs self-attention through multiple separate heads that map $\mathbf{X}$ to h different representation sub-spaces (i.e., $\{(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) \,|\, 1 \leq i \leq h\}$). The outputs of the heads are concatenated and mapped back to a

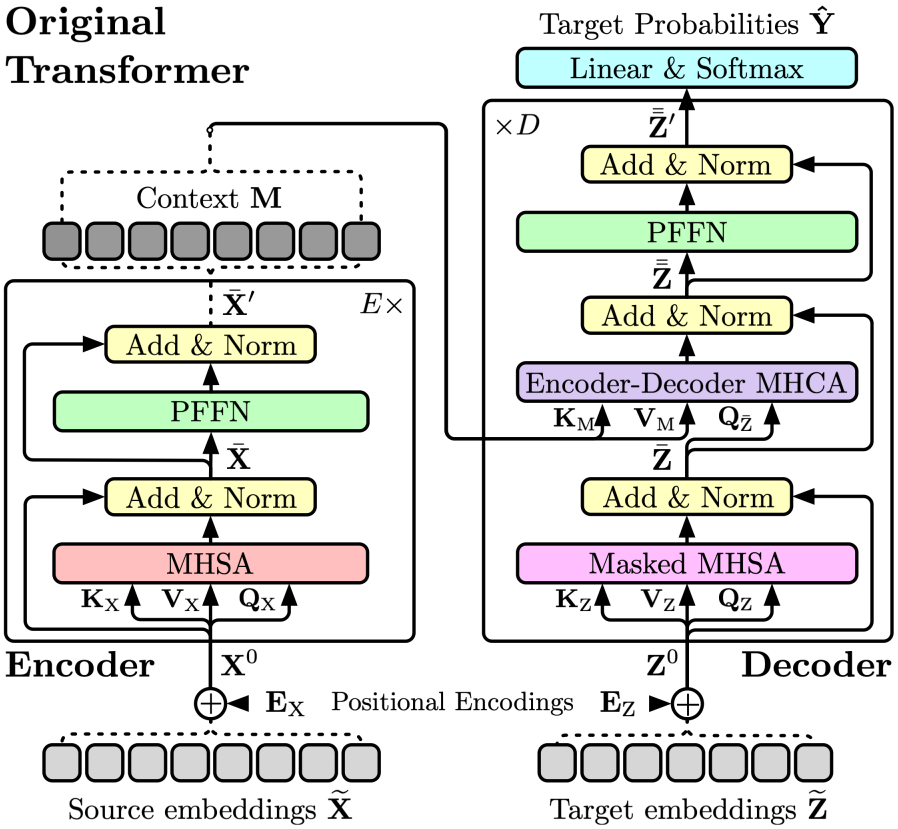


Fig. C.1: Visualization of the original Transformer proposed in [183].

d_m-dimensional space with another linear transformation $\mathbf{W}_\mathrm{O} \in \mathbb{R}^{(h \cdot d_\mathrm{v}) \times d_\mathrm{m}}$:

$$\begin{aligned} \mathrm{MHSA}(\mathbf{X}) &= \mathrm{Concat}(\mathbf{H}_1, ..., \mathbf{H}_h)\mathbf{W}_\mathrm{O}, \\ \text{where } \mathbf{H}_i &= \mathrm{Att}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i), \end{aligned} \tag{C.2}$$

where $\mathbf{H}_i \in \mathbb{R}^{N_\mathrm{x} \times d_\mathrm{v}}$ is the output of the i^{th} head, and $\mathbf{Q}_i$, $\mathbf{K}_i$, and $\mathbf{V}_i$ are computed with their own associated embedding matrices (i.e., $\mathbf{W}_{\mathrm{Q}_i} \in \mathbb{R}^{d_\mathrm{m} \times d_\mathrm{k}}$, $\mathbf{W}_{\mathrm{K}_i} \in \mathbb{R}^{d_\mathrm{m} \times d_\mathrm{k}}$, and $\mathbf{W}_{\mathrm{V}_i} \in \mathbb{R}^{d_\mathrm{m} \times d_\mathrm{v}}$ with $d_\mathrm{k} = d_\mathrm{v} = d_\mathrm{m}/h$). "Add + Norm" is then applied to come up with $\bar{\mathbf{X}} = \mathrm{LN}(\mathbf{X} + \mathrm{MHSA}(\mathbf{X}))$, where $\bar{\mathbf{X}} \in \mathbb{R}^{N_\mathrm{X} \times d_\mathrm{m}}$. After this, the following PFFN sub-layer further refines each embedding in $\bar{\mathbf{X}}$ individually (point-wise). This sub-layer is composed of two linear layers and ReLU activation function: $\mathrm{PFFN}(\bar{\mathbf{X}}) = \mathrm{ReLU}(\bar{\mathbf{X}}\mathbf{W}_{\mathrm{F1}})\mathbf{W}_{\mathrm{F2}}$, where $\mathbf{W}_{\mathrm{F1}} \in \mathbb{R}^{d_\mathrm{m} \times (4 * d_\mathrm{m})}$ and $\mathbf{W}_{\mathrm{F2}} \in \mathbb{R}^{(4 * d_\mathrm{m}) \times d_\mathrm{m}}$. Note, $\mathbf{W}_\cdot$ are independent for each layer, but we omit those indices for ease of notation. By applying this, $\bar{\mathbf{X}}' = \mathrm{LN}(\bar{\mathbf{X}} + \mathrm{PFFN}(\bar{\mathbf{X}}))$.

Decoder module. Consisting of D layers and fed with $\mathbf{Z}^0$, it substitutes MHSA with two other sub-layers. The first one, *Masked Multi-Head Self-Attention* (Masked MHSA), modifies Att in appendix 2 to include a mask, $\mathbf{B} = (b_{ij}), 1 \leq i, j \leq N_z$, impeding the access to certain tokens. This is added to the result of the dot-product in the numerator (and before the Softmax), as follows: $\mathbf{QK}^\top + \mathbf{B} \in \mathbb{R}^{N_Z \times N_Z}$, where $b_{ij} = -\infty$ iff $i < j$ (otherwise $b_{ij} = 0$). This draws attention values for the masked attention pairs to 0 when taking exponents in the Softmax. As we will see, such masking is crucial to define the auto-regressive behavior of the decoder module (avoiding tokens to attend to other tokens later in the sequence). The produced $\bar{\mathbf{Z}}$ is now passed to the *Encoder-Decoder Multi-Head Cross-Attention* (MHCA) sub-layer, which leverages the memory/context produced by the encoder, namely $\mathbf{M}$ (i.e., $\bar{\mathbf{X}}'$ at encoder's E^{th} layer), into $\bar{\mathbf{Z}}$ as follows: $\text{MHCA}(\bar{\mathbf{Z}}, \mathbf{M}) = \text{Concat}(\mathbf{J}_1, \ldots, \mathbf{J}_h)\mathbf{U}_\text{P}$, where $\mathbf{J}_i = \text{Att}(\bar{\mathbf{Z}}\mathbf{U}_{\text{Q}_i}, \mathbf{M}\mathbf{U}_{\text{K}_i}, \mathbf{M}\mathbf{U}_{\text{V}_i}) \in \mathbb{R}^{N_Z \times d_v}$ is the output of the i^{th} cross-attention head, $\mathbf{U}_{\text{Q}_i} \in \mathbb{R}^{N_Z \times d_k}$, $\mathbf{U}_{\text{K}_i} \in \mathbb{R}^{N_\text{x} \times d_k}$, $\mathbf{U}_{\text{V}_i} \in \mathbb{R}^{N_\text{x} \times d_v}$, and $\mathbf{U}_\text{P} \in \mathbb{R}^{(h \cdot d_\text{v}) \times d_\text{m}}$. Then, $\bar{\bar{\mathbf{Z}}} = \text{LN}(\bar{\mathbf{Z}} + \text{MHCA}(\bar{\mathbf{Z}}, \mathbf{M}))$. The remaining PFFN sub-layer, which is no different from the one in encoder layers, is used to produce $\bar{\bar{\mathbf{Z}}}' = \text{LN}(\bar{\bar{\mathbf{Z}}} + \text{PFFN}(\bar{\bar{\mathbf{Z}}}))$. Finally, in the D^{th} layer, the embeddings from the PFFN are each sent through a linear layer followed by softmax to generate the output probabilities over the words in the target vocabulary $\mathcal{Z}$, i.e., $\hat{\mathbf{Y}} \in \mathbb{R}^{N_z \times |\mathcal{Z}|}$.

Current Transformer trends adopted for video. Many variations to the Transformer have become common in vision and, particularly, video. First, the use of *special tokens* such as `[CLS]` (class) or `[MSK]` (mask) tokens. In video, these are parameters initialized at random and adapted during the optimization process based on the learning objective. `[CLS]` is used to condense (into a vector representation) information from the rest of token embeddings in a sequence (representing spatiotemporal patches from the video [2]), and suited for high-level tasks (such as classifying the sequence globally). Using input token embeddings instead of `[CLS]` may cause the model to be biased towards them [191]. Conversely, `[MSK]` is used to replace input embeddings and signal the Transformer to reconstruct those guided by the loss and based on the remaining tokens. This forces the Transformer to learn context from the tokens and how these relate to the masked ones. Conceived for language representation learning [32], this has been adopted also for video representation learning [178, 196].

Second, *deviations from the canonical encoder-decoder*: encoder-only or decoder-only Transformer architectures. Encoder-only are suited to produce fixed-size outputs, i.e., augmentations of the input embeddings that can be used for more granular tasks (e.g., per-frame classification) or, when used together with `[CLS]`, to come up with a global representation (e.g., sequence-level classification). For instance, [2, 5, 38] adopted an encoder-only architecture (along with the inclusion of `[CLS]`) for video classification following [34]. Instead,

decoder-only alternatives enable auto-regressive tasks if the size of the output cannot be determined a priori just by knowing the input size (e.g., to predict a series of temporal action detections). Initially proposed by [141] in NLP, these have been also followed in the context of video in [117, 174, 232]. Other trends originated in other fields have been followed: swapping the order of the residual connection and layer normalization [2, 38], although no clear general advantage of one over the other has been empirically shown yet; or replacing ReLU in the PFFN by GeLU [5, 48, 81, 189] following [32], with only [48] ablating this decision (finding out that GeLU was slightly outperforming ReLU on their task/data).

Transformer limitations. Transformers have two key limitations: first, given the pair-wise affinity computation in appendix 2, they exhibit *quadratic complexity* ($\mathcal{O}(N^2)$), which will be especially problematic for video. In appendix 4.1, we will explore some works alleviating this issue by reducing the scope of the SA operation. The second limitation is the lack of *inductive biases*. This is a double-edged sword, allowing for a general-purpose architecture that can handle any modality but severely complicates the learning process. While this can be solved through large quantities of data [34], this further adds to the computational costs of training Transformers. Throughout the three following sections, we will explore various approaches (transversal to the whole VT pipeline) to solve this issue.

3 Input pre-processing

Here, we review how video is processed before being input to the Transformer. This involves tokenization, embedding, and positioning (see fig. C.2). However, in the context of video, embedding often comes before tokenization: a separate network embeds the raw data into a continuous and compact representation, which can be used directly as a token or be further tokenized into more atomic units.

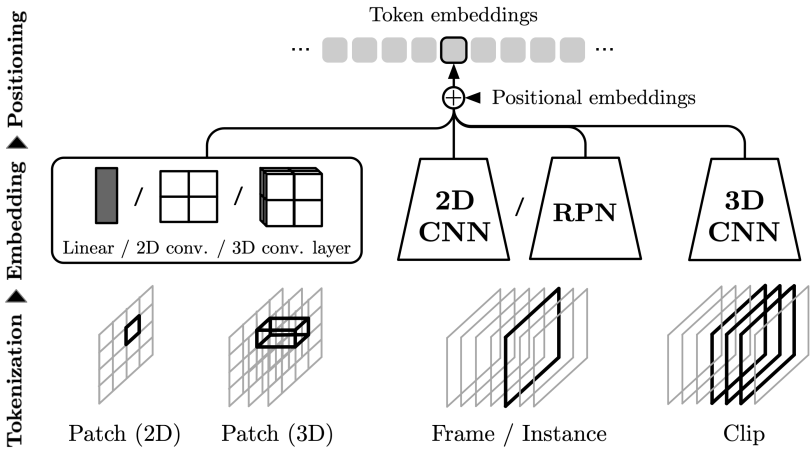


Fig. C.2: Overview of the input pre-processing step, showing tokenization and embedding strategies, as well as positioning (inclusion of positional information).

3.1 Embedding

In order to embed video, we find VTs following two main trends: *embedding networks* or *minimal embeddings*. The key difference between the two is size: while minimal embeddings are generally limited to single linear layers, large embedding networks are instantiated as full CNN architectures. Furthermore, while minimal embeddings follow the classic tokenization-then-embedding approach, full embedding networks can be used to embed full input sequences for later tokenization. In the context of video, embedding layers also function as a crucial dimensionality reduction mechanism.

Embedding network. Leveraging an embedding network (such as a CNN), can potentially ease the learning of the Transformer by providing strong initial features thanks to locality inductive biases. We can roughly categorize the choice of embedding network by the types of relationships they encode into spatial and spatiotemporal. Within *spatial embeddings*, we find 2D CNN networks, typically ResNet variants [64, 204], pre-trained on large image corpora (most commonly ImageNet [31, 150]) to learn general filters that can extract meaningful representations of individual frames. This has been shown to work effectively in the context of video [57, 67, 80, 88, 92, 98, 103, 137]. However, 2D convolutions lack the ability to model temporal information. For this reason, we also find the use of *spatiotemporal embedding* networks (e.g., in [49, 98, 137, 167, 190]). These are generally instantiated as 3D CNNs (such as I3D [16] and S3D [205]), commonly pre-trained on large video datasets such as Kinetics [14, 15] or HowTo100M [118] to produce features involving temporal relationships. Alternatively, LSTMs [106] or a hybrid ConvLSTM [162, 193, 198], can be leveraged to embed local temporal information.

While spatial embedding networks are limited to per-token spatial interactions, spatiotemporal counterparts help provide initial locally-based temporal interactions.

Minimal embeddings. Inspired by the success of ViT [34], some works [2, 5, 34, 77, 103, 219] omit deep embedding networks and subdivide the input (i.e., tokenize) and then perform embedding with only a few linear projections or convolutions. In this sense, they are guaranteed to not share information between tokens, leaving the learning of interactions between them entirely to the Transformer. Empirical studies like [5, 77], show that *stand-alone Transformers* (i.e., without complex CNN embedding networks) are as performant as CNN counterparts, although the resulting model becomes data-hungry and computationally expensive. Given that, training and deploying VTs with minimal embeddings may benefit from architectural modifications inducing necessary biases (see appendix 4).

3.2 Tokenization

When dividing a video into smaller tokens to form the input sequence to the Transformer, we find several categories depending on the token input receptive field (i.e., the extent of the original input covered by a given token before being processed by the Transformer). We distinguish between patch, instance, frame, and clip tokenization (see fig. C.2).

Patch-wise tokenization. Most VTs follow ViT [34] and employ a 2D-based patch tokenization [5, 219, 227, 234], dividing the input video frames into regions of fixed spatial size [5, 219, 234] or even multi-scale patch sizes [227]. Others propose using 3D patches (also regarded as *cubes*) instead [1, 2, 38, 109, 178], allowing to consider local motion features within the tokens themselves. While non-overlapping patches are the most common, a few works propose using overlapping 2D [103] or 3D [38] patches for smoother information flow between neighboring patches. Due to their access to neighboring information in the input, we also regard positions of intermediate feature maps from CNN embedding networks as patches (e.g., 2D in [103, 166, 186, 194] or 3D in [46, 137]), as their exact receptive field will depend on the specific setting in which they are produced. Overall, patch-based tokenization provides finer granularity, allowing to properly model spatiotemporal interactions in the VT.

Instance-wise tokenization. We refer to instances as semantically meaningful (foreground) regions that extend their reach beyond small patches but still smaller than whole frames [49, 125, 200, 239]. On the one hand, a *Region Proposal Network* (RPN in fig. C.2), such as a Faster R-CNN [149], can be used to generate region proposals and their corresponding embeddings [200]. Thus, they allow reasoning about foreground objects or region interactions. Alternatively, in [49, 95, 239], this kind of tokenization is combined with other coarser tokenizations (frame- and clip-wise tokenization) allowing to form

instance-context relationships. Instance-based tokenization can be regarded as a form of sparse sampling (e.g., [68, 152]), potentially reducing redundancy and allowing to input relatively large temporal sequences of per-frame instance representations to the VT without running into efficiency limitations.

Frame-wise tokenization. In this case, the embedding network learns initial local spatial features for each frame, and the Transformer focuses on modeling the temporal interactions among the resulting frame tokens (e.g., [94, 125, 128, 131, 189, 221, 227, 238]). This allows longer videos to be modeled (especially compared to patch tokenization), although the Transformer may have a hard time modeling fine-grained spatial interactions. However, some tasks focusing on frame-level predictions (such as video summarization [37]) may not require them.

Clip-wise tokenization. Condensing the information of several frames (clip) into each individual token allows further reducing the temporal dimension of the input (e.g., in [45, 48, 90, 167, 168, 239]). This way, the Transformer can effectively consume more frames to cover longer temporal spans. This makes clip tokenization very suitable for long-term modeling tasks. Given the high dimensionality of clips, it is necessary to embed them into single token representations through large embedding networks: for instance, [232] with C3D, [239] with 3D ResNet-50, [168] with S3D, [81] with R(2+1)D, or [90] with SlowFast, to name a few. This tokenization could also be suitable for retrieval tasks, where a high-level representation of the video is required [45, 239]. Clip-based tokenization exacerbates the pros and cons of frame-based tokenization where fine-grained information may be lost or mixed, preventing the Transformer from disentangling it later, in favor of efficiency when handling longer videos.

3.3 Positional Embeddings (PE)

Given that SA is an operation on sets, signaling positional information is necessary in order to exploit the spatiotemporal structure of videos. This is done via positional embeddings (PE), which can be either *fixed* or *learned* and then *absolute* or *relative*: fixed absolute [38, 49, 193], learned absolute [77, 90, 239], fixed relative [92, 143], or learned relative [101, 109, 197]. Absolute variants are summed to the input embeddings but can also be concatenated [77, 194, 216], while for the relative ones, the positional information is introduced directly in the multi-head attention [202].

Absolute embeddings are generally 1D. This naturally fits frame or clip tokenization to indicate position in the only remaining (temporal) dimension. However, when dealing with patch-wise tokenization, fixed 1D in raster order may seem counter-intuitive, as the last patch i-th from row j, will be regarded as closer to the first patch in the next row $j+1$, than to patch i at row $j-1$ (or $j+1$). For this reason, 2D absolute PE [46, 50] accounting for joint space

wh and time t dimensions, and 3D absolute PE [77, 193, 194, 219] for width w, height h, and t have also been proposed, disregarding [34] who found 1D learned absolute PE to suffice – at least for images.

The idea behind relative PE is that the positional information added when computing attention between token i and j depends on their relative position, making them translation equivariant. In other words, 1D relative PE added when computing attention between tokens at positions i and $j = i + k$ will be the same regardless of the value for i (i.e., $-k$). Relative PEs are generally added as an additional bias term (as in [30, 92, 107, 161]) in the dot-product between $\mathbf{Q}$ and $\mathbf{K}$ (modifying appendix 2). We find different variants of relative PEs applied to VTs, for instance [109, 174, 197] are based on decomposable attention [124], whereas [101] follows the approach of relation-aware attention [161].

3.4 Discussion on input pre-processing

Most VTs employ large CNN embeddings to reduce input dimensionality (aiding with data redundancy) and to exploit their ability to produce strong representations (thanks to local inductive biases). This significantly alleviates complexity and simplifies training when employing Transformers for video tasks. The success of these methods is clearly visible in the number of works which utilize large embedding networks as opposed to minimal embeddings. While minimal embeddings are indeed lighter than large CNN counterparts, they do result in overall more costly models if used naively. As they do not provide the necessary inductive biases, these will have to be provided elsewhere (such as in the Transformer design – see appendix 4 –, or during training, through large-scale (self-)supervised pre-training – see appendix 5). However, as we observe in appendix 6.2, this may result in better-performing models. Regarding tokenization, it has an impact on two main factors: (1) it will affect the level at which information is modeled by the VT (longer temporal spans by using frame- or clip-based tokenization, and more fine-grained spatiotemporal modeling when employing patches); (2) it will impact the input sequence length, and consequently the computational complexity of the model. For these reasons, most works use a patch-based approach accompanied by some efficient design, or frame-based tokenization, as it provides better long-term modeling scalability.

We find that the interactions between embedding and tokenization play a crucial role in defining the abstraction level and granularity at which the Transformer can model interactions. On the one hand, large embedding networks allow to produce tokens sharing information between them, guided by interactions defined by CNN's inductive biases. In this regard, it may be desirable to leverage 3D CNNs that provide local interactions among spatiotemporally neighboring positions. On the other hand, some tokenization

strategies (such as 3D patches or clips) allow the formation of fine-grained temporal interactions within the token itself. This can be further motivated by most state-of-the-art VTs employing 3D patches. In this sense, the choices of embedding network and tokenization need to be carefully considered, as they will affect the level at which spatial and temporal interactions can be formed.

Finally, the fixed absolute PEs proposed in [183] require fewer parameters than the learned counterpart. However, the latter could be learning relevant positional relations that Fourier-like approaches are unable to capture (similarly to how learned convolutional filters replaced handcrafted features). The vast majority of VTs employ these absolute variants while the use of relative counterparts is still marginal. We believe, however, that the translation equivariance these latter provide could prove useful for generalizing to unseen lengths (see appendix 7). This ability would be highly useful in the video domain as it is much more prone to display inconsistent temporal lengths (and cannot be re-scaled as easily as spatial dimensions, without harming fine-grained motion modeling – see appendix 4.4).

4 Architecture

In this section, we overview Transformer designs. The different alternatives focus on specific limitations of VTs or on better exploiting the abundant information in videos. In appendix 4.1 we analyze approaches to reduce the number of tokens accessible in a single attention operation, aiming to reduce quadratic complexity. Then, in appendix 4.2 we describe proposals to enhance the temporal modeling capabilities of VTs. Finally, in appendix 4.3 we explore specialized designs to separately capture fine-grained and coarse-level features.

4.1 Efficient designs

Given the high dimensionality of video, it may be challenging to represent long time spans without potentially incurring information loss or stumbling upon the quadratic attention matrix problem. For this reason, many works decompose full attention into multiple smaller SA. This has a two-fold benefit, as it will reduce the size of individual attention matrices while infusing different inductive biases. Two main trends are observed: (1) *restricted* approaches, which limit the scope of a single SA operation but maintain the sequence length throughout the network; and (2) *aggregation* approaches, which focus on progressively condensing information into smaller sets of tokens. A complete overview of our proposed taxonomy for efficient video designs can be seen in fig. C.4.

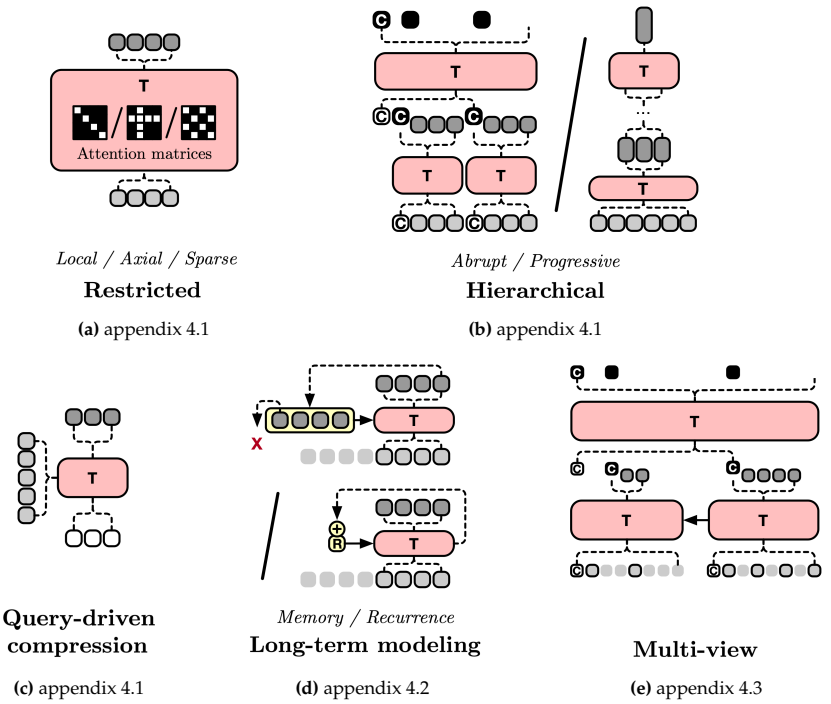


Fig. C.3: Visualization of the different design choices for VTs. Data tokens are in light gray (and black stroke if the token is used), whereas augmented tokens are in darker gray; those in white are initialized learnable tokens; and, [CLS] tokens are indicated with "C" (filled black after being augmented). Data flowing into the (T)ransformer from the side is used for cross-attention.

Restricted approaches

In order to approximate the full receptive field (i.e., the whole input sequence), restriction relies on stacking multiple smaller SA (similar to local filters in CNNs). We categorize restricted approaches by how subsets of tokens are selected for each SA. It can be by attending *local* token neighborhoods, specific *axis* (i.e., height, width or time) or *sparsely* sampled subsets of tokens (see fig. C.3a).

Local approaches are defined as the restriction by limiting attention to specific neighborhoods. Similar to *sliding* filters in CNNs, the works in [5, 27, 57, 120, 215] define the neighborhoods by sampling nearby tokens given a query. Instead, [107, 109, 197, 228] proposed limiting SA to small *fixed* windows, performing full SA separately in each of them. Relaxing the locality constraint only to time, in [9, 213] the fixed windows span all patches of a given frame. While sliding window local attention allows for more flexible learning (as each query has an independent local neighborhood), it has been shown to be cumbersome to implement [4]. Let S and T be the number of tokens in

space and time respectively (i.e., $S \cdot T = N$), local approaches reduce the computational complexity of VTs from $O((S \cdot T)^2)$ down to $O(S \cdot T)$ assuming a small (and constant) spatiotemporal neighborhood size. These approaches gain locality biases and linear complexity at the expense of non-local receptive fields, hence will require depth to account for it. For this reason, in order to allow *information to flow between windows*, we find different neighborhood sizes for each head in [57, 197], shifting the fixed windows on every layer in [107, 109] and swapping groups of features or neighborhood aggregation tokens between windows in [9, 213]. Instead, the use of global tokens is seen in [5, 228] (alternating between local and sparsely global attention), in [120] (where the `[CLS]` token attends to and is attended by all tokens, acting as a bottleneck for non-local information) and in [9] (which includes a global Transformer layer at the end).

Axial approaches define the restriction to attention by specific axes (i.e., *height*, *width*, or *time*). These can only be applied in patch-based tokenization models, where the underlying structure of the data along the different axes is kept. *Full axial attention* decomposition has been tested for VTs, either by attending over individual axes in three consecutive MHA sub-layers [5], or in a single one where each query token attends to all tokens that share with it the position in at least two axis [35]. However, it is more common to decompose attention into spatial and temporal, for modeling intra-frame and inter-frame interactions respectively. Spatiotemporal decomposition reduces computational complexity from $O(S^2 \cdot T^2)$ to $O(S^2 \cdot T + S \cdot T^2)$. The way in which spatial and temporal attention are related in the architecture will define the granularity at which spatial, temporal, and spatiotemporal interactions of the input tokens are learned. On the one hand, allowing attention to both axes at each Transformer layer allows for *spatiotemporal* relationships to form throughout layers. This can be done sequentially, through two MHSA sub-layers, as in [2, 5] (and subsequent work [146, 182, 225]) or in parallel for latter combination, seen in [2] through independent spatial and temporal heads and in [95] through separate streams for each axis. On the other hand, entirely *separating spatial from temporal* attention into consecutive modules as explored in [27, 50]. In this sense, it is not until the latter layers that temporal modeling occurs, where it may be too late for certain spatial relationships to form.

Sparse approaches. Sparse restrictions do not limit the scope of attended tokens, as opposed to local and axial approaches. Instead, given the high redundancy in video data [235], sparse models provide a way to reduce unnecessary computation while maintaining a global receptive field at each layer. Sparsity can be *embedded in the SA* operation by restricting it to fixed strided patterns for each query [5, 35]. In other words, a given query is only allowed to attend (at most) to every other token on each axis. These are generally used to complement densely local attention. Other approaches involve some form of *clustering*. This can be done through a hard assignment, where tokens get sep-

arated into groups (e.g., by k-means), allowing only attention within each of them. Intuitively, as SA contextualizes token representations through their relationships, these groupings allow attending directly to the most relevant ones for each token, discarding the ones that will contribute less. In order to allow for inter-group flow of information, [95] employs centroid SA, broadcasting contextualized cluster representations to each token within, whereas [228] uses an aggregation mechanism for later global modeling. Alternatively, in [127] **Q** and **K** are softly assigned to a subset of maximally orthogonal "prototypes" sampled from **Q** and **K**, performing SA in that reduced space.

Aggregation approaches

Aggregation-based VTs can be roughly categorized into *hierarchical* and *query-driven compression*. The key distinction is whether the input sequence length is reduced for all **Q**, **K** and **V**, or if a small set of tokens (**Q**) is used to condense information from the full input sequence (**K** and **V**).

Hierarchical designs can be further divided into abrupt or progressive. The former employ bigger neighborhoods (e.g., whole frames) and perform a single aggregation step, whereas the latter tend to work on smaller neighborhoods and involve multiple such steps (see fig. C.3b). In both cases, the improvement in efficiency comes from the fact that deeper layers will have to process a smaller sequence length. *Abrupt* approaches divide the input tokens into separate groups which are independently processed by a Transformer, to learn intra-group relationships. Then, information from each subset is aggregated, generally through a `[CLS]` token (e.g., [2, 50]), although some use learnable global pooling in the form of linear [48] or convolutional layers [154]. The aggregated representations are then fed into the next stage, modeling inter-group relationships. We only find one work leveraging pure temporal hierarchy [48], which models frame-then-clip interactions. It is more common to employ spatiotemporal hierarchical models. These works ([2, 9, 50, 73, 120, 217, 228]) are the aggregation equivalent of spatiotemporal axial methods: a first module (generally a ViT [34] or Swin [108] architecture), learns spatial patch-wise interactions, and a second one models frame-level temporal interactions. Interestingly, in [154] multiple aggregation tokens are used for each frame, containing different features. As we discuss later in appendix 4.4, these approaches may lose the ability to model fine-grained features after aggregation, potentially missing relevant temporal cues.

Progressive approaches, tackle this limitation by learning spatiotemporal interactions at all levels. In works such as Video Swin [109] and MViT [38] (as well as their followups [68, 97, 107, 185, 187, 196, 201]) non-local interactions are learned at each level, whereas in [93] the first layers are limited to local interactions. In both cases, sequence length is progressively aggregated by local neighborhoods (i.e., through learnable local pooling) while expanding

the tokens' dimensionality. While this increased model capacity for deeper layers will require more parameters (weight matrices **W** quadratically grow with the number of feature channels), it is generally compensated by smaller dimensionality in shallower layers. The work of [228] combines both types of hierarchy, by progressively downsampling in the spatial module, for latter aggregation and high-level temporal modeling.

Query-driven compression. Another aggregation-based approach consists in defining the set of queries **Q**, such that $N_Q \ll N$. Then, the computations are reduced from $O(N^2)$ to $O(N \cdot N_Q)$. In these, SA is performed only on the tokens that correspond to Q, while K and V will be attended over via cross-attention. With this, the N_Q queries will iteratively access the whole input to distill the most useful information and aggregate it in the token embeddings corresponding to the queries. The intuition behind this is similar to how the input tokens to the decoder get refined by repeatedly cross-attending to the encoder's memory **M** (see fig. C.1). These queries are either an aggregated or sub-sampled version of the input data, or they are an independent set of tokens. *Aggregating* the input into queries (e.g., through global pooling) can be used to build global streams while maintaining access to a broader low-level context within K and V. This may be useful for tasks that require a high-level representation of the input clip (e.g., video retrieval [48], scene or action classification [157] or group activity recognition [95]). Interestingly, in [166] this idea is developed by forming a reduced set of queries at each layer. In particular, T and S embeddings resulting from spatial and temporal average pooling respectively, are concatenated and used to attend the full set of keys and values. Alternatively, a *sub-sampled* version of the input can be used to reason about specific regions or objects (e.g., by extracting a small set of boxes from the input clip to be used as queries [49, 237]). Using a fixed set of *learnable queries* to cross-attend the input was first explored in [77] to build a global stream, where latent embeddings are used to progressively gather information from the raw high-dimensional input. In VT literature it is more common to use these learnable queries in an object-centric fashion, extending on DETR [12] (used to detect objects at each frame) and propagating detection tokens to build recurrent Transformers (e.g., [116, 236]). Alternatively, a set of independent *text-based queries* can be defined from the text modality to aggregate relevant visual information for video question answering [86]. This idea naturally extends the original Transformer, replacing the textual encoder with a video one while maintaining the auto-regressive text decoder, for video captioning [74, 92, 94, 117] or dense captioning [74, 221, 238] (through further event sampling).

4.2 Long-term (temporal) modeling

Capturing long-term dynamics might be crucial for video tasks, as events observed at a given moment could potentially be only understood by looking far away in time. We here focus on works that propose dealing with long-term temporal modeling. We roughly categorize them into memory- (e.g., [92, 201]) and recurrence-based approaches (e.g., [116, 212]). Whereas recurrent ones aggregate information into fixed-size representations, memory-based ones are variable-size and allow selective attention. In both, portions (i.e., frames/clips) of the videos are processed sequentially in a sliding window fashion to keep manageable compute and GPU memory but still ensure relevant information from past windows is within reach.

Memory

Naively caching many past raw (high-dimensional) input frames quickly becomes prohibitive. Instead, one can store global frame features [39, 206] or convolutional maps late in the embedding network [215], intermediate embeddings across Transformer layers (e.g., those from patches [201]), or the Transformer's output embeddings [7]. In particular, when dealing with patch embeddings, aggregation might be needed before storing them [201]. On top of that, some works maintain several memories with different temporal reach (long/short) [206, 215], abstraction level [201], or granularity (fine/-coarse) [7, 201].

Memories are mostly *accessed* via either cross-attention [39, 206, 215] or self-attention [7, 201]. By concatenating input and memory tokens sequence-wise to perform self-attention, the cost of the operation is $O((N_M + N_X)^2)$. Although manageable with small memories, cross-attention turns out to be much more affordable, with cost $O(N_M \cdot N_X)$ if we assume $N_X \ll N_M$. Either way, if N_M happens to be too large, one can *reduce the number of tokens* on-the-fly when accessing them [39, 201, 206] by either query-driven compression [39, 206] or progressive aggregation [201] – both seen in appendix 4.1. On the one hand, memories leveraging query-driven compression follow a two-stage bottleneck compression: a first Transformer compresses the memory into a smaller set of tokens, whereas a second one "decompresses" the output of the former into a larger set but still much smaller than the original memory. In the case of [206] the second Transformer is also deeper than the one in the first stage. It also uses two separate sets of learnable tokens to perform the aggregation in both stages, while [39] uses a hard selection of memory tokens in the first stage (obtained via *Farthest Point Sampling* [138]). Besides the efficiency gained from such two-stage factorizations, we intuit differentiated underlying roles of the Transformers. While the first focuses on rough selection/compression, the second tries to recover as much information as possible, aggregating and further refining

embeddings. On the other hand, progressive memory aggregation throughout the Transformer layers provides later access to finer-to-coarser details. For instance, [201] keeps spatially aggregated $K^{\ell}_{(t-M^{\ell}):(t-2)}$ from previous timesteps after a learnable pooling and concatenates with lastly cached memory that is to be compressed in this iteration (i.e., K^{ℓ}_{t-1}), and the current input's K^{ℓ}_{t} to be used in the ℓ-th MHSA sub-layer (and analogously for V embeddings).

Multiple memories (e.g., short- and long-term) can be separately accessed and their respective memory-enhanced embeddings fused [215] or, alternatively, a short-term memory (with fewer tokens) can drive the compression of the long-term one [206]. In multi-layer memories [201], the ones in later Transformer layers implicitly access information provided by earlier ones, effectively allowing local memory accesses to approximate the full receptive field of the memory in deeper layers. As we move forward in time, memories are *discarded* in a First-in First-out (FIFO) fashion [39, 201, 206, 215]. A notable exception is [7], which leverages the self-attention weights to discard memory token(s) that are less attended by the rest.

Recurrence

Drawing inspiration from RNNs/LSTMs, recurrence mechanisms have also been proposed to deal with long video sequences. Here we distinguish between recurrence applied between intermediate layers in the VT [92, 212] and outside of it [116, 236].

Within the first category, we find RViT [212] and MART [92]. RViT [212] is essentially a ViT-like spatial Transformer that propagates the output of every self-attention sub-layer forward in time. Acting as recurrent states, these are added to the embeddings from the current time step after projecting both to its own **Q**, **K**, and **V**. Instead, MART [92] leverages the embeddings alone to form **Q** whereas a sequence-wise concatenation of those with the recurrent state is used to derive **K** and **V**. Differently from RViT, the recurrent state is not the output of SA, but the result of a gating mechanism between the previous state and the current input embedding.

Recurrence can also be established outside the Video Transformer. In other words, the output embeddings from the Transformer at time $t-1$, namely $\hat{X}^{D}_{t-1}$, can be propagated to its own input at t. In the context of object detection, [116, 236] propose an encoder-decoder architecture where the decoder augments a set of learnable tokens while attending to the encoder's representation of the current frame. At time $t = 0$, the decoder augments an initial set of learnable tokens that will become recurrent tokens. At $t > 0$, the decoder augments the sequence-wise concatenation of the recurrent tokens at $t-1$ and added learnable tokens at t to capture newly appeared objects. Trained using pairs of frames, these can still deal with long sequences during inference. One may argue that having a token for each object could be regarded as a

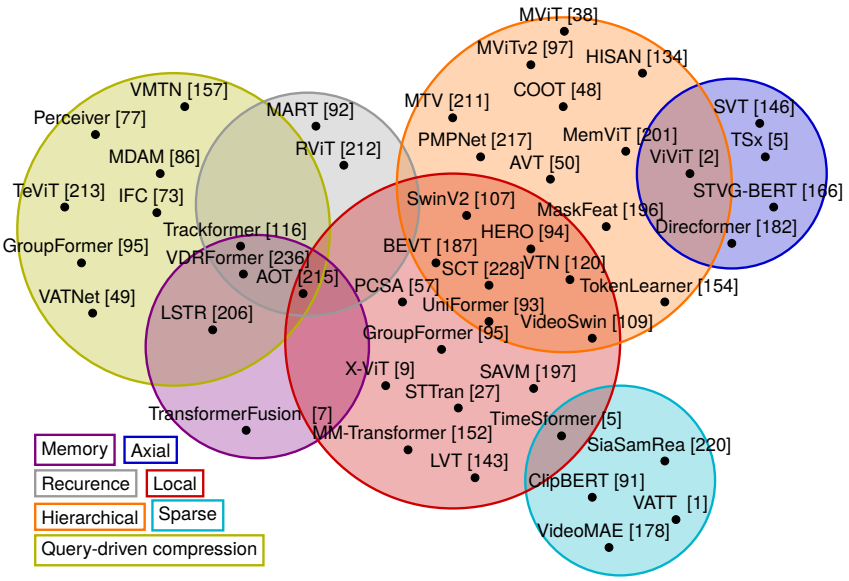


Fig. C.4: Venn diagram displaying our proposed taxonomy of efficient VT designs (best viewed in color). We describe Local, Axial and Sparse approaches in appendix 4.1, Hierarchical and Query-driven compression in appendix 4.1, and Memory and Recurrence in appendix 4.2.

form of memory, but from the point of view of time, the information is being recurrently aggregated into a fixed-size representation.

4.3 Multi-view approaches

Opposed to dense sampling of single views, a few VTs define multiple views of a given video to solve the task at hand in a cooperative fashion. Instance-based contrastive approaches instead employ multiple views to drive the loss (see appendix 5.2). Note that multi-view approaches are related to multi-view sampling at inference (see appendix 6.1), but crucially, the former leverage this technique also during training. A clear example of this parallel is [91], which defines *sparse views* by uniformly sampling video frames with a fixed stride but varying starting positions. Then, separate streams process each view and the final classification is reached by averaging predictions in a late fusion manner. This work could be seen as the sparse equivalent to fixed window local restriction. In this sense, it only incurs $O(R^2k)$, where k is the number of sparse sequences (i.e., $R \cdot k = N$). As weights are shared across streams, no parameter increase is incurred.

Interestingly, many approaches define views by *varying the resolution* of a given clip, while allowing interactions between them to form throughout the

network (i.e., early fusion). This was first explored for video in [227] by using patches of different spatial size at each head, and later extended to time in [211] by using 3D patches instead. In the latter case, a multi-stream network is used where each stream models the same video but tokenizes with different temporal resolution (inspired by the SlowFast Network [41]), allowing information flow between views through cross-attention and a final global stream (in an abrupt hierarchical fashion). In [198] a similar architectural setting is used, but the views are sampled from the output of progressively deeper layers of a ConvLSTM embedding network. In this sense, each view holds a smaller spatial resolution, but a bigger temporal context. Intuitively, these methods use redundancy to their advantage, helping the network become robust to missing information in single views, while each stream models a coherent representation of the full input.

4.4 Discussion on Architecture

VT designs focus on reducing computational complexity and handling the redundancy of videos without compromising spatiotemporal modeling capabilities. Furthermore, restrictions imposed on VTs to make them more efficient will bias them towards favoring certain kinds of relationships. For instance, abrupt hierarchy learns temporal translation equivariance in spatial layers by modeling each frame independently, and local approaches that enforce locality biases.

However, efficient designs and inductive biases do not handle redundancy. Video redundancy can be mostly attributed to appearance-based semantics varying slowly through time, even when small variations in specific pixels occur [235]. However, the extended information provided by these subtle changes in many consecutive frames may be crucial to properly model fine-grained motion features [41]. In order to learn spatiotemporal relationships from video, this must be taken into account. Reducing spatial redundancy may be desirable, as it will allow focusing on more relevant parts of the video (e.g., through aggregation or sub-sampling of tokens). However, this requires careful consideration: removing certain information too early into the network may limit the formation of crucial temporal interactions later on. Prior works on modeling video with CNNs have shown this to be the case: early aggregation of spatial features hinders the formation of fine-grained motion features [99, 159, 179], and temporal pooling seems to hurt spatiotemporal representation learning [17, 40]. With Transformers, tackling this may involve taking into account non-local neighborhoods before deciding which information is to be discarded.

Motivated by this, we derive three crucial aspects for spatiotemporal modeling: (1) explicit spatial redundancy reduction while (2) allowing to model temporal features at all levels in (3) high-fidelity temporal contexts. Different

VTs exhibit varying degrees of capabilities in these three aspects. *Restricted approaches* allow for low-level temporal modeling and, due to the lack of aggregation, always maintain temporal fidelity. Given their potential to overlap low- and high-level features they can be suitable for both low-level (e.g., segmentation [35]) and high-level tasks (e.g., classification [5]), but with certain limitations. Hierarchical approaches effectively exhibit (1) and (3) through aggregation on the spatial dimensions only (except in [93]). Particularly, for *progressive hierarchy* (e.g., [97, 107]), the gradual increase in channel dimensionality provides deeper layers with a larger capacity to represent high-level concepts while further limiting the modeling of redundant low-level features. Furthermore, by leveraging different levels of spatiotemporal non-local contexts (e.g., [38, 109]) at least in deeper layers (e.g, [93]), they guarantee that extended temporal fidelity is exploited before aggregation. In contrast, the *abrupt counterparts* (e.g., [50]) may be suffering from early aggregation. While training end-to-end may infuse temporal feedback into spatial layers (which may be sufficient for appearance-biased video benchmarks, see appendix 5.3), they may lack proper motion modeling. This can be addressed by allowing to form spatiotemporal interactions before aggregation, either locally (by explicitly sharing information between neighborhoods [9, 73, 213, 228] as well as by using 3D patches [2, 211]) or globally [154]. *Query-driven compression* approaches reduce redundancy through aggregation when used to derive global streams [48, 95, 157], or through sparsity when reasoning of individual objects or regions [49, 95]. In both cases, the small set of queries forms high-level representations of (parts of) the input, while maintaining temporal fidelity in keys and values. However, they may exhibit a limited capability to form low-level temporal features. While iterative accesses may alleviate the dangers of early aggregation for high-level tasks (e.g., classification [49, 77, 157]), low-level tasks may require to also evolve the fine-grained input representations [95] or to infuse them back with high-level features from the queries [166] (similar to clustering-based sparse approaches). This is similar to the behavior exhibited by *recurrent* approaches. As temporal information is collapsed into the recurrent state, they may suffer from early aggregation, which may be especially detrimental for high-level tasks [212]. However, these approaches may excel on applications that only require low-level reasoning of the current observation, enhanced with the forwarded high-level past context (such as for tracking [116], segmentation [215] or dense video captioning [92]). *Memory*-based approaches exhibit great capabilities for preserving the temporal resolution of the input. They can tackle redundancy through aggregation (e.g., upon storing [201] or dynamically on access [206]) or sparsity (either by storing only some past observations [215], by dropping elements in the memory according to their relevance [7] or only attending to a small subset of memory tokens [39]). Finally, *multi-view* approaches working at different input resolutions explicitly allow the formation of separate coarse- and fine-

level features while allowing interactions between them [198, 227]. However, as redundancy is not explicitly managed, the success of these methods may be limited to computationally heavy models [211]. Sparse counterparts heavily downsample the input sequence, hurting temporal fidelity and requiring to compensate with other modalities [1, 91, 220].

5 Training a Transformer

The two main limitations of Transformers will heavily influence the way in which they are trained. On the one hand, large-scale pre-training aids Transformers to overcome their lack of inductive biases [22, 32, 34], but recent studies suggest that self-supervised pre-training (see appendix 5.2) alleviates the need for large supervised datasets [178, 196]. On the other hand, some solutions to the lack of inductive biases aggravate computational costs. CNN embedding networks add to the memory footprint and potentially overflow GPU memory when training, especially if done end-to-end. Avoiding overfitting big models requires strong regularization [208] and lots of data [229], which is further problematic when handling several stages of training that require more time and compute. Finally, leveraging self-supervised tasks is computationally heavy, especially for video.

5.1 Training regime

We next explore how VTs are trained, from a lens of embedding networks and pre-training. Pre-training involves one or more training stages before transferring the network to a downstream task, for which the model is either fine-tuned or linearly probed (training a few linear layers on top of the frozen Transformer).

End-to-End training with minimal embeddings. End-to-end training of deep neural networks has proven to outperform multiple-stage algorithms. To ease memory limitations while allowing for end-to-end training of the Transformer, it is common to use minimal embeddings. Some train in a supervised fashion [2, 5, 38, 77, 197, 228], directly for a downstream task on large datasets, such as Kinetics-700 [14], or ImageNet21K [150]. However, all these leveraged efficient architectures and thanks to the inductive biases these designs provide, the network will pick up on relevant patterns faster, and more capacity can be given to Transformer layers. Other works aiming for smaller datasets train aided by some data augmentation [198, 219, 223] or self-supervised losses [1, 50, 103, 117, 227] on medium to large datasets. Stand-alone Transformers seem to be able to learn without large CNN embeddings if aided by the inductive biases that efficient designs, data augmentation, or self-supervised losses provide. Still, most of these require multiple stages of training either through large datasets or computationally heavy self-supervised techniques.

End-to-End with embedding networks. Other works train Transformer and deep CNN embedding layers end-to-end either with a pre-trained embedding network [23, 86, 166], fine-tuning just the later layers [157, 167], or training end-to-end from scratch [111, 129, 186, 193]. Some were able to train end-to-end by capping Transformers to 1~4 layers [81, 217, 234], suggesting that just a few Transformer layers after a large embedding network may be enough to boost performance. Some others' success is attributable to leveraging efficient designs (e.g., local SA [57]) or weight sharing [90] – that reduces the effective number of parameters to be stored in memory (discussed later in appendix 7). Finally, [49, 195] report having substantial computational resources available, which allowed them to fit in memory both, a large embedding network and a big Transformer. Empirical studies on both image [29, 144] and video [93] Transformers have consistently found improvements when training Transformers and CNN embedding layers end-to-end. This may further be seen in works reporting improved CNN-based results alone after being trained as the embedding net of a Transformer [90, 167, 169], pointing towards CNNs benefiting from long-term temporal feedback provided by the Transformer layers.

Frozen embedding networks. The most common approach by far for VTs is leveraging some large pre-trained and frozen CNN embedding network. These are then used for feature extraction, which further boosts cost-effectiveness, as they can be pre-computed. Transformer layers are then trained for a downstream task on those features. Compared to end-to-end training from scratch, it is often cheaper and more efficient to employ SOTA models, which have been carefully tuned to perform well on some supervised task. While it is definitely common to use medium to large datasets (as in [74, 75, 92, 125, 131, 221, 232, 238]), with this approach, many video works [10, 11, 25, 46, 98, 134, 136, 137, 174, 189] are still able to train the Transformer on small datasets (<10k training samples). Nevertheless, these approaches are limited by the quality of the pre-trained features and could be biased towards the task they were trained on (which is generally supervised).

Pre-trained Transformers. Video-based pre-training has proven to work best for video classification tasks [185, 196], maybe due to the distribution gap, as pre-training only on images does not provide any motion cues. Nevertheless, image-based pre-training may provide stronger spatial features, given the higher variability of appearance and number of categories (providing better semantics regarding objects) compared to video datasets (where many consecutive frames contain similar appearance statistics). It is for this reason that we find many VTs leveraging image pre-trained Transformers, commonly on some ImageNet variant [31, 150]. This is generally done in one of two fashions. On the one hand, some works [2, 9, 50, 120, 211, 215] leverage a pre-trained image Transformer (generally ViT [34] or Swin [108]) as the spatial stage of an abrupt hierarchical VT, training the later temporal layers from scratch. On the other

hand, a pre-trained image Transformer can be directly adapted by using 3D patches to factor time in (as well as inflating linear embeddings and positional biases to account for this change) before fine-tuning for video [38, 107, 109, 196]. Finally, some object-centric approaches (e.g., [116, 194, 195, 237]) leverage pre-trained Transformer-based object detectors (e.g., DETR [12]) as initialization.

5.2 Self-supervised pretext tasks

Harvesting large annotated datasets incurs additional labeling costs, and may further influence towards human-induced annotation biases [26, 151]. *Self-supervised learning* (SSL) has been recently shown to alleviate data needs for an equivalent supervision-based pre-training (e.g., [178, 196]), while providing more robust [66] and general features [24, 56, 84]. Despite the great success of SSL in both NLP [32] and Image Transformers [61], they are not as widespread in the video domain, which could be attributed to the large costs involved in such a process. Therefore, we next analyze the benefits and limitations of SSL for VTs, so as to motivate further research in this area.

Traditional time-related pretext tasks (see [156] for a complete review) are rarely used in the context of VTs. They are generally limited to shuffling the input sequence, and training the network to correctly reorder it [58, 94, 182, 225], effectively learning coherent temporal dynamics. However, these have not found as much success [156] compared to (1) *Instance-based learning* and (2) *Masked Token Modeling* (MTM), which we explore next. The former learns sequence-level representations that are invariant to different spatiotemporal perturbations, whereas the latter mask individual token representations of the input and tries to reconstruct them.

Instance-based learning

Instance-based approaches for VTs leverage contrastive losses (generally *InfoNCE* [121]) to make representations of whole sequences invariant to certain augmentations. These approaches define one anchor $\mathbf{x}$, a positive sample $\mathbf{x}^+$ and a set of G negative samples to contrast against $\{\mathbf{x}_g^-\}$, where $1 \leq g \leq G$. These tasks force representations for the positive pair to be similar, while it drives apart representations for the negative (dissimilar) pairs. Minimizing InfoNCE can be seen as maximizing a lower bound on the mutual information between $\mathbf{x}$ and $\mathbf{x}^+$ [121]. These losses have also been used in the context of cross-modal matching for VTs, as explored in [153]. Hence, we only briefly discuss them in the context of video retrieval in the Supplementary and focus here on their uses for video only.

View mining. Positive pairs tend to be differently augmented versions (generally regarded as *views*) of the same sample. In VTs (and generally for video), it is customary to apply spatial augmentations (e.g., random cropping, color jitter-

ing, horizontal flips, or Gaussian blur) consistently through time (i.e., applying the same augmentation to all frames [139]). By aligning multiple views' representations, the model learns to be invariant to such perturbations. However, spatial augmentations alone are not enough for video SSL [156], and generating temporal views needs to be done carefully. For instance, reversing or randomly shuffling a clip may make the model invariant to temporal causality. In VTs (similar to other video literature [43]), it is common to use multiple temporal [58, 169, 185, 200] or spatiotemporal [129, 146] crops of a given video to form the positive pairs, with varying temporal spans [129, 146, 185] and frame-rates (i.e, speed) [146, 185], whereas negatives are sampled among the rest of training videos. Learning invariance to such changes may be useful for high-level tasks where a wide abstract understanding of the video is enough. Nevertheless, this could disregard local view-dependent information in favor of redundant cross-view information [135], favoring the formation of appearance-biased features (see appendix 5.3). In order to tackle this, some VTs use multiple global and local potentially overlapping views as positives [129, 146, 185], which may allow for better modeling of part-whole relationships. Intuitively, this forces global views to preserve information in the local ones, while maintaining global context awareness in local views. Alternatively, the alignment task can be relaxed, skewing away from learning absolute invariance to changes between views. One example is seen in [169], which conditions alignment on the temporal shift between crops. Another example is seen in VT works introducing asymmetries in the networks computing the different views' representations: using additional predictors [185], momentum encoders [146, 185] (originally proposed in [62], are believed to behave as network ensembles [13]), and even CNNs [58] (probably helping infuse some locality bias from CNN representations into the Transformer). Introducing some of these asymmetries has indeed been found to boost downstream performance on image [177] and video tasks [43]. Intuitively, they may be relaxing the alignment task into a more predictive setting, allowing features to be aware of context, not so much invariant to it.

Negative sampling. One crucial limitation of contrastive approaches is their need for large negative sets [21]. These are generally mined from the batch, which can be very limiting in the context of full video representations, as it may not always be possible to hold enough different instances in a batch. VTs tackle this through large memory banks that store representations of past batches [104, 160, 185] (which may further serve as regularizers, due to storing sample representations from past iterations produced by the same model with slightly different weights) or through hard negative mining (forcing the network to learn small nuances in the views by trying to separate somewhat similar samples, measured by feature representation distances [90, 96]). Finally, we also find works dropping negatives altogether. One example is seen in [146], which formulates learning as instance-based classification, where

every positive view has to be classified in the same pseudo-class. Another example is the work in [220], where multiple sparse views of the input are independently processed and the aggregated prediction is used to distill the consensus into single view streams.

Masked Token Modeling

MTM draws inspiration from the *Masked Token Prediction* task proposed in BERT [32]. It randomly replaces some input tokens with a learnable `[MSK]` token and the network is trained to predict (classify) the replaced tokens. This forces the Transformer to learn contextualized representations of the input. However, different from language tokens, visual tokens cannot be easily mapped to a discrete and limited-size vocabulary so as to pose MTM as a classification task. For perspective, a pixel codebook would require $255^3 \approx 16$M distinct elements, whereas BERT employed a vocabulary of 30K. Furthermore, posing it as a classification task would disregard the distance of the prediction to the actual ground truth value, distracting the network with high-frequency details of the data which could be irrelevant. To solve this issue in the context of VTs, we roughly find three families of approaches, categorized by the type of target: (1) working at *feature* level either through regression [18, 94, 96, 196] or contrasting [25, 90, 94, 167], as well as (2) *quantization* of visual tokens [168, 187]. Interestingly, some works have actually found success (3) regressing the original token in *pixel* space [107, 178]. We also find VTs classify token contents [200, 239], but as these require manual annotations, we do not delve into them here.

Feature-based MTM works regress a feature-based representation of the masked tokens. This can be posed as a prediction (e.g., using an MSE loss) [18, 94, 96] or as a contrastive task [25, 90, 94, 203]. The target token representation is obtained from the input embedding network (e.g., [90, 94, 203]) or from an external encoder [167]. In this sense, by requiring an additional network, these models potentially incur additional compute and memory costs. In order to avert this, [196] proposes using the HOG features of the masked region, which are cheaper to compute and can be pre-computed. Interestingly, the work of [50] uses causal masked SA instead of replacing tokens with `[MSK]`. In this sense, all tokens are tasked with solving feature-based MTM by trying to predict the next token representation (in a predictive coding setting [121, 147]).

Quantization-based MTM involves discretizing video tokens to a limited codebook, generally requiring some pre-trained network to define it. For instance, in [168] an S3D [205] followed by hierarchical k-means is used for both embedding the tokens prior to the Transformer and the discrete (cluster assignation) pseudo label for the prediction, whereas in [187] a VQ-VAE [145] is used instead, but only to generate the ground truth for the masked tokens. The use of quantization makes it possible for these models to optimize the

network with a classification objective, akin to NLP counterparts. Similar to many feature-based MTM, however, these approaches also require an additional pre-trained network.

Pixel-based MTM directly regresses the pixel space for masked regions [107, 178]. They do not require any further networks or computing additional features, making them very simple to implement. However, pixels as targets have been argued to focus on irrelevant high-frequency details of data, which could be detrimental for high-level tasks [60]. Nevertheless, this may be more nuanced and require further research, as we discuss next in appendix 5.3.

5.3 Discussion on training strategies

Training stand-alone VTs requires balancing solutions to the lack of inductive biases with potentially limited computational budgets. This implies factoring in large datasets, SSL, and efficient designs while accounting for the large dimensionality of videos, properly sized clips, batches, and architectures. VTs are dominated by fully supervised training aided by *large frozen CNN embeddings* (which are not so common in other fields, such as NLP), and disregard pre-training of Transformer layers. On the one hand, long-range temporal interactions provided by Transformers boost CNN's performance in many application scenarios [11, 46, 74, 88, 101, 125, 134, 137, 174]. On the other hand, the embedding network provides initial representations and dimensionality reduction, alleviating Transformers' training limitations. Nevertheless, this approach caps the potential of Transformers to model spatiotemporal interactions and depends on the transferability of the pre-trained embedding features (e.g., distribution or task shift).

The canonical pre-training then fine-tuning paradigm acts as a *smart form of initialization*. Skewing from it may allow avoiding catastrophic forgetting [115] while achieving more generalizable features. For example, by incorporating self-supervised auxiliary losses during fine-tuning, as done by some VTs [50, 225]. In [94] a training schedule is proposed that samples a different (self-)supervised task at each batch, showing improved results for video retrieval as more tasks are added. Alternatively, recent works (e.g., [51, 187, 230]) deviate from the trend of image-based pre-training and achieve promising results by optimizing for image and video tasks in a joint manner.

SSL is not as widespread for VTs when compared to supervised or image-based initialization. However, we believe VTs could greatly benefit from large-scale unlabeled videos, as well as from the inductive biases SSL provides. In this sense, we see great promise in the current developments on SSL that are better suited to train visual Transformers. MTM could be seen from the lens of generative-based pre-training as it bears great resemblance with CNN-based inpainting [126]. We believe that the success of MTM may be attributable to Transformers providing explicit granularity through tokenization. In order to

conquer the complex global task of inpainting large missing areas of the input, MTM *divides* it into smaller local predictions. This is especially true in both 2D- and 3D-based patch tokenization approaches [178, 187, 196]. Intuitively, the model needs an understanding of both global appearance and motion semantics as well as low-level local patterns to properly gather the necessary context to solve token-wise predictions. This may allow VTs to learn more holistic representations (i.e., better learning of part-whole relationships). Nevertheless, given the high redundancy of videos it could be trivial for the network to find shortcuts, borrowing information from neighboring spatiotemporal positions instead. It has been found that high masking ratios (e.g., 40%-60% in MaskFeat [196] or even 75%-90% in VideoMAE [178]), especially compared to NLP (15%-20% in BERT [32]) or images (20%-50% in MAE [61]), indeed force the network to capitalize on global relationships of the data, as seen by improved performance on high-level semantic tasks (see appendix 6.2). Furthermore, ablations in [178, 196] suggest that the masking strategy can also impact the learning of such shortcuts, showing that masking blocks of tokens in space consistently through time helps to avoid them. Regarding the choice of target for MTM, quantized and feature-based seem to work best for video [196] (albeit requiring an additional pre-trained network). Pixel-based provide the cheapest target but are generally discarded arguing they may fixate on irrelevant high-frequency details of data. However, the generally used MSE loss has been shown to disregard such details [123, 126, 233], so further research may be needed. Finally, we highlight HOG features, which provide the best compute/performance trade-off (see appendix 6.2), as they are cheap to compute while providing partial invariance to various deformations.

Despite requiring large batches for negative mining, instance-based contrastive approaches have consistently shown potential for high-level video tasks [156]. By contrasting differently spatiotemporal augmented views, the network learns invariance to appearance perturbations, spatial scale, and occlusions, as well as changes of perspective or illumination naturally present in video [54, 192]. However, the model may also become invariant to temporal translation and deformation, effectively disregarding fine-grained motion dynamics and biasing it towards appearance-based static cues (which is enough for appearance-biased datasets (e.g., UCF101 or Kinetics) where the presence of certain objects may suffice to predict an action class [8, 71]). As we discussed in appendix 5.2, re-introducing motion modeling requires relaxing the alignment task through network asymmetries (e.g., [58, 146]) or careful sampling techniques (e.g., [129, 185]) to balance part-whole relationship learning. However, compared to MTM, it is easier for these approaches to overlook low-level view-dependent temporal information, crucial for proper motion modeling [139, 222].

In this context, we see promise in combining instance-based contrastive learning and MTM, both in multi-task scenarios [167, 169, 200] as well as

feature-based contrastive MTM [90, 94, 167, 203] (as opposed to direct regression). These latter could combine the holistic feature learning of MTM while potentially accounting for the uncertainty of modeling missing information through contrastive losses (as the model is not tasked with explicit hard prediction [60]). For instance, in [94], this alternative is found to outperform L2 feature regression in the context of video-moment retrieval. These approaches remain, to the best of our knowledge, unexplored in the context of patch-based tokenization, where the cardinality of the negative set would be much larger than for instance-based approaches (allowing for many hard negatives from the same sequence as well as easy negatives from all other sequences in the batch). Nevertheless, it is still unclear what these models are actually learning, so future research is needed for proper interpretation of SSL features, which currently are mostly evaluated based on their success on downstream performance [79, 156].

6 Performance on video classification

The task of video classification has attracted the most research in Transformers for video, given the generality of the task and availability of large datasets for training and evaluation, things that allow for more comprehensive performance analysis. Next, we overview the particularities of video classification (appendix 6.1) and then analyze VT state-of-the-art performance on it (appendix 6.2).

6.1 Video classification

Video classification aims to predict the class of a given input sequence of frames. For the task, a VT will encode descriptive high-level global representations of a given sample. Then, some linear layers followed by a softmax provide a class-score probability distribution. The category with maximum probability should match the ground-truth class. VTs competing to become state-of-the-art in classification tend to be standalone (i.e., use minimal embedding), and thus will be the ones we cover. Next, we present the benchmarking datasets, experimental protocols, and details on the sampling of the clips.

Evaluation datasets. The most popular dataset is the large-scale *Kinetics-400* (K400) [15], consisting of 306K 10-second clips and 400 manually annotated human actions categories with at least 400 examples per class. Kinetics-600 (K600) – an extension of K400 with 495K clips and 600 classes – is only used for pre-training, but not for evaluation. K400/K600 are however known to be appearance-biased [225]. To better assess the modeling of more complex temporal dynamics, most works are also evaluated on *Something-Something v2* (SSv2) [55, 113]. SSv2 is an egocentric human action dataset where some of the categories can only be distinguished by having an understanding of the arrow

of time (e.g., "Moving [sth] away from [sth]" vs "Moving [sth] closer to [sth]"). SSv2 consists of 220K videos of duration ranging from 2 to 6 seconds and 174 fine-grained categories.

Experimental protocols. We find two learning protocols being followed: training from scratch or pre-training the model first. *Training from scratch* is rarer because of the size of the models (especially, their larger variants). When following *pre-training*, the weights learned during the first stage are used to initialize the model that is to be trained in the downstream dataset/task. Common pre-training strategies for video classification are (a) image-based pre-training on ImageNet, and either (b) supervised or (c) a self-supervised video pre-training (generally on video datasets larger than the downstream one, e.g., K600 for evaluating K400 and K400/K600 for SSv2). After initialization, the models are trained on the downstream dataset, fine-tuning existing weights and adapting new ones.

Clip sampling. Models are fed with trimmed video clips. These are relatively short, with a number of frames T' typically 8 to 64 frames and fixed spatial resolution $S' = H' \times W'$ pixels (often $H' = W' = 224$, hence shortened to $S' = 224^2$, see "Input" in tables C.1 and C.2). However, to make sense of these numbers, and especially T', it is crucial to consider the temporal stride τ, i.e., the step between clip frames when sampling them from the original video. A larger τ extends the temporal span of the clip w.r.t. the video without incurring extra computation costs, while also reducing the redundancy among otherwise nearby sampled frames. For instance, with $\tau = 4$ and $T' = 64$, a clip covers a temporal span equivalent to a densely ($\tau = 1$) sampled clip of 256 frames. Importantly, τ must be chosen factoring in the temporal resolution of videos (e.g., ~25 FPS in K400) and the fine-grained motion information one is willing to sacrifice in favor of context.

Views. The clips generated can be regarded as *temporal views* (related to the views described in appendix 5.2, which are used for some methods during pre-training). During training, one temporal view per video is gathered at a random temporal position. These are constructed with fixed size $T' \times S'$ and stride τ. For inference, most models follow a multi-view approach: the classification decision for the video is achieved by averaging the prediction obtained from different spatiotemporal crop views.

6.2 Comparison among state-of-the-art models

To draw comparisons we consider the factors defined by the columns of tables C.1 and C.2. Among those, the most interesting one to study is perhaps the pre-training strategy, which will drive the rest of the section, separately analyzing K400 and SSv2.

K400: training from scratch. Doing so, we only find MViTv2 [97] and its predecessor MViT [38]. The main difference between the two is the inclusion of an

extra residual pooling connection and the use of relative positional encoding. With these, "MViTv2-B 32@3" (82.9%) performs better than its older counterpart "MViT-B 32@3" by +2.7%. In fact, it also surpasses "MViT-B 64@4" – which has an increased temporal receptive field (2.6x) – by +1.7%. Later in [196], the same authors explored different initialization strategies and showed overfitting of the larger variants of MViTv2 when not using effective initialization. This can be seen for "MViTv2-L↑", with an increased spatial resolution (from 224 to 312) and compute (from 51 MP to 218 MP), performing worse (-0.7%) than "MViTv2-B 32@3'. Although this is to be expected, the smaller variants are still able to learn from scratch successfully – as we will see later, even better than 3D ConvNets. Given the need for pre-training of larger models, we next discuss the two most popular strategies in the context of K400 and demonstrate its large positive effect (e.g., "MViTv2↑ 32@3" boosts its results from 82.2% to 85.3% by leveraging image-based pre-training).

K400: image pre-training. The majority of VTs pre-train on either ImageNet-1K ("IN"), ImageNet-21K ("IN21"), or JFT-300M ("JFT"). IN and IN21 consist of 1K and 21K classes and over 1.2M and 14M examples respectively, whereas JFT is a non-public dataset with 300M multi-label images and 18,291 non-mutually-exclusive labels. Other works have been using their own image datasets or extending public ones. For instance, "Video-SwinV2-G" [107] (86.8%), being the best performing model, extended IN21 (14M images) with a private dataset of images ("P" in Tab. C.1), totaling 70M samples. Close second is "MViTv2-L↑ 40@3" [97] (86.1%), with weights pre-trained exclusively on IN21 while only dropping by -0.7% with respect to the first one, but with 14x fewer parameters. The third is "MTV-H" [211] (85.8%), this one pre-trained on JFT with 300M images. Unfortunately, in this work, the authors used JFT to pre-train their largest models ("MTV-L" and "MTV-H") and IN21 to train "MTV-B"/"MTV-B (320)", therefore not validating the actual contribution of JFT w.r.t. IN-21K for any of the variants; making difficult to discern the actual contribution of the model scaling. Also, TokenLearner [154] completely relies on JFT for all the experiments, with its best model "TokenLearner 16at18 (L/10)" (85.4%) coming fourth. It was ViViT [2] that showed how the same model variant trained on JFT, "ViViT-L (JFT)" (83.5%), was considerably improving upon the same variant pre-trained on IN-21K ("ViViT-L"), by +1.8%. It is, hence, of great merit that "MViTv2-L↑ 40@3" (86.1%) still surpasses, respectively by +0.3% and +0.7%, the results of "MTV-H" and "TokenLearner 16at18 (L/10)". It is true that compared to those, MViTv2 variant utilizes larger spatial (312^2, versus 224^2 and 256^2 pixels) and temporal receptive field (120 vs 64 frames), but the number of TFLOPs and the amount of pre-training data to process are still both lower: 14 MP versus 300 MP in JFT for MTV and TokenLearner, and 42 TFLOPs versus 47 and 48 TFLOPs.

In terms of cost-effectiveness, we find "UniFormer-B" [93] (83.0%), "SCT-L" (83.0%) [228], "Direcformer" [182] (82.8%) – this one based on [5]-, and

"MViTv2-S" (82.6%) [196]. These models only suffer a drop between -3.1% and -3.3% of accuracy but between 10x and 70x less FLOPs w.r.t. "MViTv2-L↑ 40@3".

K400: video (self-supervised) pre-training. An emerging trend in the literature is to perform all SSL pre-training, fine-tuning and evaluation on the same dataset [178, 185, 187, 196]. "MaskFeat-L↑ 40@3" [196] reaches 86.4%, thus showing the contribution of MaskFeat (SSL) pre-training compared with supervised training on the same architecture, i.e., MViTv2, by +0.3%. That result of MaskFeat is also only -0.2% behind the best image-based pre-trained model (i.e., "Video-SwinV2-G"). Then, "MaskFeat-L↑↑ 40@30" by switching K400 with K600 and slightly increasing the spatial resolution from 312 to 352 (still lower than 382 of "Video-SwinV2-G"), the model obtains state-of-the-art results (87%), outperforming any of the image pre-trainings. VideoMAE [178] comes second in this category consisting of a ViT backbone with 3D inflation of the patch embeddings. This outperforms all image-based pre-trained models, except for "Video-SwinV2-G". Thus it seems learning motion priors during pre-training has a very positive effect on performance when targeting video classification.

K400: ConvNets. For the sake of completeness, we compare VTs to 3D ConvNets, which were state-of-the-art right until VTs managed to surpass them. See how "MViTv2-S" (81.1%) trained from scratch, exceeds the performance of comparable ConvNets: "SlowFast R101+NL" (79.8%) and "X3D-XXL" (80.4%). This might be attributable to the higher temporal fidelity of MViTv2 being more profitable than extra context – at least on short videos. The number of views for testing was also higher for both (30 versus 5 in " MViTv2-S"). Nonetheless, it also consumes 18x - 22x fewer TFLOPs and works on a smaller spatial resolution (224 only, versus 256 or 312). Switching to "MViTv2-B 32@4" (82.9%), we see how trained from scratch this model does better than ConvNets pre-trained on the very-large weakly-annotated video dataset IG65 (i.e., 'R(2+1)D-152" [47] (81.3%) and "ir-CSN-152" [180] (82.6%)), even when using half the views.

Moving to the study of SSv2, we found none of the works train from scratch. Another thing to note is the number of temporal views used because of the shorter duration of SSv2 videos compared to Kinetics. Despite that, the temporal dynamics are harder to capture as we will see next.

SSv2: image pre-training. Although less common than for K400, there are works that pre-train on image datasets. Among the ones using IN, "DirecFormer" [182] (64.9%) is the one performing the best. It surpasses its own backbone ("TS" [5]) in all the variants by forcing the learning of temporal order of shuffled input frames via auxiliary SSL. "TIME" [225] is another one using auxiliary SSL ablated with different VT backbones. This one, not so much competing in performance with larger model variants, still points out the effectiveness of temporal guidance for image-based pre-trained models

when transferred to the downstream video task. Finally, trained on IN-21K, "X-ViT" (66.4%) [9] is the absolute winner in this category. Unfortunately, by focusing on efficiency alone, it is not able to compete with heavier models that are supervisedly pre-trained on K400.

SSv2: video (supervised) pre-training. It is quite common to reuse supervisedly trained checkpoints on Kinetics by transferring them to SSv2 for fine-tuning. These have often also been pre-trained on IN-1K/IN-21 before Kinetics. However, to better disentangle video and image contributions, we focus first on video-only pre-training models, and concretely on those relying on K600. Looking at "MViT-B 32@3" (67.8% pre-trained on K600) and "MViT-B 64@4" (67.7% pre-trained on K400), with a temporal receptive field of 96 to 128 frames respectively, we see there is no improvement in SSv2 by extending temporal context, but slightly better performance when keeping finer temporal resolution (stride 3 instead of 4). Even more interesting is that the deeper "MViT-B-24 32@3" (with 24 layers) outperforms +0.9% upon the 12-layered 32@3 variant. This suggests more complex temporal dynamics might require not necessarily increasing temporal resolution, but higher abstract spatiotemporal semantics being modeled. That or advancements in architectural designs to better model those without going deeper, as done by "MViTv2-B 32@3" (70.5%) also with 12 layers. Finally, if we have a look at models that leverage image-based pre-training before Kinetics, we find further improvement (e.g., "MViTv2-B 32@3" from 70.5% to 72.1%). What seems to be not as useful, according to "UniFormer" variants, is to switch from K400 to K600.

SSv2: video (self-supervised) pre-training. The only model pre-training on SSv2 is VideoMAE ("VMAE"), which turns out to be the best performing one. In particular, "VMAE (ViT-L) 32@2" (75.3%) slightly improves upon "MaskFeat-L↑ 40@3", those being self-supervisedly pre-trained on K400 (74.4%) or K600 (75.0%). It does so with almost half the temporal context, half the FLOPs, and – importantly – with fewer data. All in all, VideoMAE and MaskFeat seem to point out pixel- and feature-based MTM approaches compare favorably with "SVT" (instance-based invariance learning) or "BEVT" (quantization-based MTM) despite the latter also using image-based pre-training.

6.3 Discussion on performance

We have introduced the task of video classification and analyzed the performance of state-of-the-art models on Kinetics-400 and Something-Something v2. Our main finding was that the pre-training strategy was the biggest factor influencing the performance of VTs for video classification, thus the following discussion will address three questions related to this: (1) *Can Video Transformers be trained from scratch?*, (2) *Which is the best pre-training strategy?*, and (3) *How can we effectively model stronger spatiotemporal dynamics?*.

For the smallest models, *training from scratch* seems to be doable. In particular, MViT [38] and MViTv2 [97] are able to, respectively, compete with and slightly surpass 3D ConvNets trained from scratch. In fact, MViTv2 even outperforms those pre-trained on very large weakly-annotated video datasets (e.g., IG-65M). In particular, we attribute the success of those to the locality bias they infused (via the local pooling-based progressive aggregation discussed in appendix 4.1), which allows these models to go deeper without exploding in computational complexity while still keeping their self-attention operation global. However, training from scratch seems to be the least desirable strategy to follow.

Among pre-training strategies, video-based ones, either supervised (e.g., on K400/K600 before fine-tuning SSv2) or self-supervised, are superior to image-based pre-training alone. Image-based supervised pre-trained models seem to be able to partially compensate the lack of temporal modeling with appearance diversity by leveraging huge – often non-public – image datasets (e.g., JFT [2, 154, 211] or extensions of IN21 [107]). Alternatively, image-based self-supervised learning only competes with video pre-training when leveraging prohibitively large models [107]. However, parting from learned very diverse and general appearance features will not harm the modeling of time in later stages, but serve as a good initialization for subsequent video-based pre-training (e.g., all those works that combine IN/IN21 and K400/K600 before fine-tuning on SSv2) or fine-tuning stages with temporal SSL auxiliary losses (e.g., [182, 225]). On the other hand, we can see how self-supervised video pre-training surpasses supervised regimes. In particular, MaskFeat [196] (with MViTv2 [97] backbone) and VideoMAE [178] (with a plain ViT [2]) outperform those pre-trained on video in a supervised way.

For the successful modeling of spatiotemporal patterns, Masked Token Modeling stands out (see appendix 5.3). Concretely, MaskFeat (feature-based MTM) obtains the best results on K400 and is second best on SSv2, which is dominated by VideoMAE (pixel-based MTM). Interestingly, these models do not require extra data or manual annotations to surpass all other models, being able to self-supervisedly pre-train on the evaluation dataset itself. Unfortunately, instance-based invariance learning (e.g., [146, 185]), being that popular for image representation learning, heavily underperforms compared to MTM for video classification.

Apart from the importance of pre-training, other findings in appendix 6.2 we might want to highlight are: first, that the modeling of the complex spatiotemporal dynamics seems to benefit more from deeper models and temporal granularity than extended temporal spans; second, that naive adoptions of image and NLP models (e.g., VTN [120], which leverages the image-based ViT [4] to model space and the language-based Longformer [4] to model time) might not work that well; and, third, that although joint self-supervised learning on image and video (i.e., BEVT [187]) is promising, it still has a long way to go.

	Pre-train	Name	Ref.	Input	TF × v_t × v_s	MP.	Acc.
ConvNets	-	SlowFast (R101+NL)	[42]	16@8 × 256^2	0,23 × 10 × 3	60	79,8
		X3D-XXL	[40]	16@5 × 312^2	0,19 × 10 × 3	20	80,4
	IG65 (video)	R(2+1)D-152	[47]	32@1 × 128^2	0,25 × 10 × 1	118	81,3
		ir-CSN-152	[180]	32@2 × 224^2	0,10 × 10 × 3	NA	**82,6**
Scratch	-	MViT-S	[38]	16@4 × 224^2	0,03 × 5 × 1	26	76,0
		MViT-B		32@3 × 224^2	0,17 × 5 × 1	37	80,2
		MViT-B		64@4 × 224^2	0,46 × 3 × 3	37	81,2
		MViTv2-S	[97]	16@4 × 224^2	0,06 × 5 × 1	35	81,0
		MViTv2-B		32@3 × 224^2	0,23 × 5 × 1	51	**82,9**
		MViTv2-L↑	[196]	32@3 × 312^2	2,06 × 5 × 3	218	82,2
Image pre-tr. (I)	IN	UniFormer-B	[93]	16@4 × 224^2	0,10 × 4 × 1	50	82,0
		UniFormer-B		32@4 × 224^2	0,26 × 4 × 3	50	83,0
		Swin-B	[109]	32@2 × 224^2	0,28 × 4 × 3	88	80,6
	IN21	SCT-L	[228]	24@10 × 224^2	0,34 × 4 × 3	60	83,0
		Swin-B	[109]	32@2 × 224^2	0,28 × 4 × 3	88	82,7
		Swin-L↑		32@2 × 384^2	2,11 × 10 × 5	200	84,9
		TS	[5]	8@16 × 224^2	0,20 × 1 × 3	121	78,0
		ViViT-L-FE	[2]	32@2 × 224^2	3,98 × 1 × 3	352	81,7
		VTN-3 (Aug)	[120]	250@1 × 224^2	4,22 × 1 × 1	114	79,8
		DirecFormer	[182]	8@32 × 224^2	0,20 × 1 × 3	124	82,8
		Mformer	[127]	96@3 × 224^2	0,96 × 10 × 3	NA	81,1
		Mformer↑		64@4 × 336^2	1,19 × 10 × 3	NA	80,2
		X-ViT	[9]	16@1 × 224^2	0,28 × 1 × 3	92	80,2
		X-ViT		16@1 × 224^2	0,28 × 2 × 3	92	80,7
		MTV-B	[211]	32@2 × 224^2	0,4 × 4 × 3	310	81,8
		MTV-B↑		32@2 × 320^2	0,96 × 4 × 3	310	82,4
		RViT-XL	[212]	64@NA × 224^2	11,90 × 3 × 3	108	81,5
		MViTv2-S	[196]	16@4 × 224^2	0,07 × 10 × 1	36	82,6
		MViTv2-L↑		32@3 × 312^2	2,06 × 5 × 3	218	85,3
		MViTv2-L↑	[97]	40@3 × 312^2	2,83 × 5 × 3	218	86,1
	(IN-21 + P) (SSL)	SwinV2-G↑	[107]	8@NA × 384^2	NA × 4 × 3	3 K	**86,8**
	JFT	ViViT-L-FE	[2]	32@2 × 224^2	3,98 × 1 × 3	352	83,5
		ViViT-H		32@2 × 224^2	3,98 × 4 × 3	352	84,9
		MTV-L	[211]	32@2 × 224^2	1,50 × 4 × 3	NA	84,3
		MTV-H		32@2 × 224^2	3,71 × 4 × 3	NA	85,8
		TokenLearner	[154]	64@1 × 256^2	4,08 × 4 × 3	450	85,4
Video pre-tr. (V)	K400 (SSL)	LSTCL (Swin-B*)	[185]	16@8 × 224^2	0,36 × 5 × 1	88	81,5
		MaskFeat-S	[196]	16@4 × 224^2	0,07 × 10 × 1	36	82,2
		MaskFeat-L↑		32@3 × 312^2	2,06 × 5 × 3	218	86,3
		MaskFeat-L↑		40@3 × 312^2	2,83 × 4 × 3	218	86,4
		MaskFeat-L↑↑		40@3 × 352^2	3,79 × 4 × 3	218	86,7
		VideoMAE (ViT-B)	[178]	16@4 × 224^2	0,18 × 5 × 3	87	80,9
		VideoMAE (ViT-L)		16@4 × 224^2	0,60 × 5 × 3	305	84,7
		VideoMAE↑ (ViT-L)		32@4 × 320^2	3,96 × 5 × 3	305	85,8
	K600 (SSL)	MaskFeat-L	[196]	16@4 × 224^2	0,34 × 10 × 1	218	85,1
		MaskFeat-L↑↑		40@3 × 352^2	3,79 × 4 × 3	218	**87,0**
I+V	IN + K400 (SSL)	SVT (TS)	[146]	8@NA × 224^2 + 64@NA × 96^2	0,20 × 1 × 3	121	78,1
	IN (SSL) + K400 (SSL)	BEVT	[187]	16@NA × 224^2	0,28 × 4 × 3	88	80,6
		BEVT (Dall-E tknzr.)		16@NA × 224^2	0,28 × 4 × 3	88	81,1

↑: increased spatial resolution.
"IN21 + P": extension of IN21 with a private private dataset (70M images in total).

Table C.1: Accuracy (top-1) on Kinetics-400. "Input": temporal and spatial size of the views; "TF": TFLOPs; v_t and v_s: the number of temporal and spatial views; "MP": parameters ($\times 10^6$); and "Pre-train": pre-training strategy.

	Pre-train	Name	Ref.	Input	TF × v_t × v_s	MP.	Acc.
CN	IN	TDN (R101)	[42]	8@1 × 256^2 + 16@1 × 256^2	0,2 × 1 × 3	198	**69,6**
Image pre-tr. (I)	IN	TS*	[225]	8@NA × 224^2	NA × 1 × 3	121	62,1
		Mformer*		8@NA × 224^2	NA × 1 × 3	NA	63,8
		TIME (TS*)		8@NA × 224^2	NA × 1 × 3	121	63,7
		TIME (Mformer*)		8@NA × 224^2	NA × 1 × 3	NA	64,7
		DirecFormer	[182]	8@32 × 224^2	0,20 × 1 × 3	124	64,9
	IN21	TS	[5]	8@16 × 224^2	0,20 × 1 × 3	121	59,5
		TS-HR		16@16 × 448^2	1,70 × 1 × 3	121	62,2
		TS-L		96@4 × 224^2	2,38 × 1 × 3	121	62,4
		ViViT-L	[2]	32@2 × 224^2	3,98 × 1 × 3	352	**65,9**
		X-ViT	[9]	16@NA × 224^2	0,28 × 1 × 3	92	66,2
		X-ViT		32@NA × 224^2	0,42 × 1 × 3	92	66,4
Video pre-tr. (V)	K400	MViT-B	[38]	32@3 × 224^2	0,17 × 1 × 3	37	67,1
		MViT-B		64@4 × 224^2	0,46 × 1 × 3	37	67,7
		MViTv2-B	[97]	32@3 × 224^2	0,23 × 1 × 3	51	70,5
	K600	MViT-B	[38]	32@3 × 224^2	0,17 × 1 × 3	37	67,8
		MViT-B-24		32@3 × 224^2	0,24 × 1 × 3	53	68,7
	K400 (SSL)	LSTCL (Swin-B*)	[185]	16@8 × 224^2	0,36 × 5 × 1	88	67,0
		MaskFeat-L↑	[196]	40@3 × 312^2	2,83 × 4 × 3	218	74,4
	K600 (SSL)	MaskFeat-L↑	[196]	40@3 × 312^2	2,83 × 1 × 3	218	75,0
	SSv2 (SSL)	VideoMAE (ViT-B)	[178]	16@2 × 224^2	0,18 × 2 × 3	87	70,6
		VideoMAE (ViT-L)		16@2 × 224^2	0,60 × 2 × 3	305	74,2
		VideoMAE (ViT-L)		32@2 × 320^2	1,44 × 1 × 3	305	**75,3**
Image + video pre-tr. (I + V)	IN + K400	UniFormer-B	[93]	16@4 × 224^2	96,67 × 1 × 3	50	70,4
		UniFormer-B		32@4 × 224^2	259,00 × 1 × 3	50	71,2
	IN21 + K400	Swin-B	[109]	32@2 × 224^2	0,28 × 1 × 3	88	69,6
		X-ViT	[9]	16@1 × 224^2	0,28 × 1 × 3	92	67,2
		MViTv2-B	[97]	32@3 × 224^2	0,23 × 1 × 3	51	72,1
		MViTv2-L↑		40@3 × 312^2	2,83 × 1 × 3	218	**73,3**
		Mformer	[127]	96@3 × 224^2	0,96 × 1 × 3	NA	67,1
		Mformer↑		64@4 × 336^2	1,19 × 1 × 3	NA	68,1
		MTV-B	[211]	32@2 × 224^2	0,40 × 4 × 3	310	67,6
		MTV-B↑		32@2 × 320^2	0,96 × 4 × 3	310	68,5
		RViT-XL	[212]	64@NA × 224^2	35,70 × 1 × 3	108	67,9
		MViTv2-S	[97]	16@4 × 224^2	0,06 × 1 × 3	35	68,2
		ORViT MF-L	[68]	32@4 × NA	1,26 × 1 × 3	148	69,5
	IN + K600	UniFormer-B	[93]	16@4 × 224^2	96,67 × 1 × 3	50	70,2
		UniFormer-B		32@4 × 224^2	259,00 × 1 × 3	50	71,2
	IN21 + K600	SCT-L	[228]	24@10 × 224^2	0,34 × 4 × 3	60	68,1
	IN + K400 (SSL)	SVT (TS)	[146]	8@NA × 224^2 + 64@NA × 96^2	0,20 × 1 × 3	121	59,2
	IN21 + K400 (SSL)	MaskFeat-L	[196]	40@3 × 224^2	2,83 × 1 × 3	218	**73,3**
	IN (SSL) + K400 (SSL)	BEVT	[187]	16@NA × 224^2	0,32 × 1 × 3	88	70,6
		BEVT (Dall-E tknzr.)		16@NA × 224^2	0,32 × 1 × 3	88	71,4

*: re-implementation.
↑: increased spatial resolution.

Table C.2: Accuracy (top-1) in Something-Something v2. See caption in table C.1.

7 Final Discussion

In this survey, we have comprehensively analyzed trends and advances in leveraging Transformers to model video.

Complexity. Given the inherent complexity of Transformers and the great dimensionality of videos, most changes focus on handling the computational burden. This is done transversally across the various stages of the VT pipeline. We find this is most generally addressed with frozen embedding networks, easing Transformer learning through the provided inductive biases and reducing input dimensionality. The Transformer in this context is used to enhance these representations through long-range interactions, which seems enough to boost performance in many areas of application. However, this trend alone may be limiting the potential of Transformers to learn non-local low-level motion cues. We are excited to see novel VT designs (e.g., MViT [38]) which greatly reduce complexity thanks to the inductive biases embedded in the Transformer itself (sometimes becoming lighter than CNN counterparts, see appendix 6.2). We also see great promise in MTM when separating the representation learning from the reconstruction which is done by an additional decoder discarded after pre-training [178]. This separation allows the (deeper) encoder to only leverage unmasked tokens, which greatly alleviates training complexity when using large masking ratios. Crucially, this sacrifices the possibility to leverage certain designs for the VT, as the input structure is lost (e.g., local or hierarchical approaches may not find enough tokens in a given neighborhood to learn valuable representations).

Spatial redundancy and temporal fidelity. Modeling temporal interactions requires special considerations not present when only modeling appearance (i.e., with image Transformers). On the one hand, the highly redundant appearance information in videos [178, 235] makes it difficult to model information-rich representations that avoid repeatedly representing similar or same sub-representations. It has been proven that pure attentional models lose expressivity with depth, collapsing towards uniform attention in deeper layers [33, 34, 77]. It further seems that this smoothing of the attention matrix is accompanied by highly uniform token representations and even redundant weight matrices [20]. Proper handling of video redundancy is crucial in VTs, where we hypothesize these observations may get exacerbated. On the other hand, few exceptions aside, many current designs and SSL approaches directly inherit from image approaches without careful consideration of the nuances that come with time, making them strongly biased to learn appearance features. As we have seen, allowing temporal features to form at both low- and high-level while accounting for the necessary temporal fidelity is also critical. In this sense, reducing redundancy for video should mostly target appearance features.

Key advancements on VTs. Regarding *architectural choices*, we find progressive hierarchical approaches to stand out. They carefully consider non-local temporal contexts before spatial aggregation. This effectively tackles the redundancy problem while avoiding early aggregation problems that hinder the learning of fine-grained motion features. However, to properly handle long-range interactions without losing temporal fidelity, memory-based approaches with adequate sampling or aggregation techniques may be crucial. Regarding *self-supervised learning*, MTM forces to leverage global spatiotemporal semantic contexts through high masking ratios when solving local token-wise predictions. By doing so, it is driven to learn both motion and appearance cues necessary to solve the task. Nevertheless, we look forward to further developments in sampling techniques for instance-based contrastive approaches that skew from appearance biases toward motion-specific features.
Inductive biases. As we have seen, inductive biases are a pivotal aspect for all aspects of VTs. They alleviate the need for data by providing stronger cues for the Transformer to pick up faster. Frozen *embedding networks* could be regarded as infusing task-specific biases, as the Transformer is bounded to learn on the provided representations, which in turn are dependant on the pre-training auxiliary task. Some examples include detected bounding boxes of objects [50, 68], higher-level (action) features [239], or scene, motion, OCR, and facial features, among others [45]. We have also seen how most *architectural designs* infuse some inductive biases to aid in training the Transformer. However, in this regard, VT literature so far is limited when considering infusing motion-specific biases that help the network to pick up relevant spatiotemporal cues. Just two works deviate from this trend. Motionformer [127] proposes trajectory attention to reason about aggregated object or region representations through implicit motion paths in both time and space. Differently, OrViT [68] leverages separate motion and appearance streams. The former learns trajectories of individual objects or regions that later get added to patch-wise token representations of the whole video appearance, effectively infusing motion into it. Finally, besides locality biases or invariance to perturbations induced by different *training losses*, we deem it interesting to highlight works infusing causality biases by training the network to sort shuffled video sequences [182, 225]. Furthermore, the work in [58] combines the benefits of both CNNs and Transformers for video learning through a siamese distillation setting, effectively inducing CNN locality biases into the Transformer.

7.1 Generalization

It has been shown that vision Transformers are robust to various perturbations [6, 114], suggesting they may be better able to form abstract semantic representations [231], probably due to their ability to leverage non-local contexts [130]. These findings point towards Transformers favoring out-of-

distribution (OOD) generalization [65]. Few VTs have studied this on OOD data [103, 106, 128, 160, 198, 239] or evaluated the learned features in other settings [48, 168, 174, 219], showing consistent results. Nevertheless, the issue of generalization of video may entail studying other aspects that are still under-researched. For instance, we hypothesize that generalizing to varied frame sampling rates may require further training or conditioning the network on said rates such that it may become robust. We observed, however, that some existing work may display capabilities to generalize to unseen sequence lengths.

Unseen sequence length. One issue to account for when processing sequences of unseen length is positional encodings. While we expect them to generalize to shorter sequences, they may have trouble when dealing with longer ones (which may be desirable to provide extended temporal fidelity during deployment), as no positional information is present to account for them. We find few VTs showing that PEs can easily be extended by fine-tuning the model on longer sequence lengths [1, 38]. Recent VT works have also seen promising results when leveraging input conditioned RPEs [93] or by learning a small network that computes log-scale relative positional biases [107]. These advances pose a great potential to easily generalize to unseen sequence lengths. Similarly, long-range modeling architectures could also handle sequences of any given length, as they process inputs sequentially within fixed windows, but they may require RPEs [201].

Multi-modality. Video is inherently multi-modal (i.e., contains visual and auditory information), which could be leveraged to learn more general representations. The lack of inductive biases makes Transformers very versatile tools to handle any modality. It has been found that high-level semantic features learned by language-based Transformers generalize to other modalities [110, 170]. In the context of VTs we find VideoBERT [168], where a pre-trained language BERT [32] model is used as initialization, showing promising results in this direction. Lately, there has been a great interest to use these architectures to solve multi-modal tasks [153]. We hypothesize that the lack of inductive biases may allow Transformers to learn shared multi-modal representation spaces that exhibit better generalization capabilities. When targeting video-only tasks (e.g., tracking, segmentation, classification) we see potential in multi-modal SSL to learn such spaces. We find a few VTs leveraging instance-based multi-modal learning approaches [48, 90, 94, 167] to align representations from various modalities. For instance, [1] successfully performs heavy downsampling of video by aligning it with audio and textual modalities; or the model in [129] which learns to attend to the spatial sources of audio within the video by aligning audio with visual crops. Interestingly, this alignment is further enforced in some works by sharing weights between Transformer streams modeling different modalities [1], sometimes even showing improved results compared to not sharing [90]. As pointed out in [156], this has proven

to be very useful for video (at least in the context of classification) outside of Transformers, especially when pairing video with audio or text.

7.2 Future work

VTs are still in their infancy and despite seeing clear trends, much more research is needed. First of all, we find a severe lack of explainability tools that properly assess the kind of spatiotemporal representations that different designs and self-supervised losses provide. Overlaying head-specific attention heat-maps of the first layer over a given input may provide some ad-hoc explanations on what the model deems relevant [78, 158, 199]. Even if some VTs have explored this direction (e.g., [7, 50, 77, 94, 120, 125, 128, 129]), this technique may prove overly cumbersome for video, as it requires inspecting such per-sample activations for multiple full video sequences. Possible future venues could analyze the learned patterns of attention preferred by different heads (as in [119]), which may clue on relevant design choices that favor such patterns. Besides, the aforementioned versatility of Transformers could be used to probe the model through textual descriptions (as done for images in [176]). Furthermore, we see an interesting future direction in analyzing whether video-based features would also generalize to other modalities. For instance by following a similar approach as in [170] and tuning a few adapter layers to map other modalities into the video representation space. Beyond current MTM approaches, other traditional losses could be adapted to the token granularity, such as 3D jigsaw puzzles [85]. Regarding instance-based methods, adapting recent developments to images such as Barlow Twins [226] or VicReg [3] which focus on preserving view-dependent information, may prove beneficial to video modeling. Nevertheless, further research is still needed to alleviate the computational burden of self-supervision in video. Finally, key advancements in architectural choices and training techniques for VTs are mostly limited to high-level tasks, hindering analysis of the contributions they provide for general video representation learning. Furthermore, VTs have barely tackled generative tasks such as frame prediction [143, 197] or inpainting [103, 227]. We believe that token granularity and long-range modeling capabilities of Transformers could benefit these tasks. However, given the high complexity, they entail and the tendency of Transformers to disregard high-frequency details may pose severe challenges. We hope our contributions in this paper will entice further research in many different areas of application and boost our current understanding of Video Transformers.

Acknowledgments

This work was funded by the Pioneer Centre for AI, DNRF grant number P1, by the Spanish project PID2019-105093GB-I00, by ICREA under the ICREA Academia program, and by Milestone Research Programme at Aalborg University.

8 Appendix

The supplementary material includes the following: the general table with a general overview of the most relevant Video Transformers surveyed in appendix 8 and details about specific Transformer trends for different video tasks in a more application-oriented manner in appendix 8.

General table

The general table overviews the most relevant Video Transformers surveyed. Note that due to its length, the table has been split into two subtables, table C.3 and table C.4.

Task-specific designs

In this section, four major subsections review specific designs of the following tasks: Action classification in appendix 8.1, Video translation (e.g., captioning) in appendix 8.2, Retrieval in appendix 8.3, and Object-centric tasks (e.g., detection and tracking) in appendix 8.4. This is followed by short summary subsections regarding the remaining tasks: Low-level in appendix 8.5, Segmentation in appendix 8.6, Summarization in appendix 8.7, and Others in appendix 8.8.

8.1 Classification

Regarding video classification, few works rely on pure Transformers [1, 2, 5, 109, 228] that for the most part focus on efficiency: both [2] and [5] test various space-time decompositions, whereas [2] also tests tokenization strategies (2D vs 3D patches). They found that a pre-trained ViT [34] encoding 2D patches with a temporal encoder on top performed the best. The works of [109] and [228] propose different types of restricted attention: the former restricts locally in shifting windows and the latter by only attending to previous frame's patches after having exchanged information with another efficient attention mechanism [87]. In [38] they opt for 3D patches whose receptive field

	Name	Ref.	Yr.	Architecture				Input				Train.
				Arch.	Aggr.	Restr.	Long-t.	Backbone	Embd.	Tknz.	Pos.	SSL
Classification	TimeSformer	[5]	'21	E	-	LAS	-	-	Minimal Embedding	P	LA	-
	PE	[90]	'21	E	-	-	-	-	SlowFast [42], RN-50 [64]	C	LA	P
	CBT	[167]	'19	E	-	-	-	-	S3D [205]	C	-	P
	ViViT	[2]	'21	E	H	A	-	ViT [34]	Minimal Embedding	P	LA	-
	ELR	[137]	'19	E	-	-	-	-	I3D [16]	P	-	-
	FAST	[219]	'21	E	-	-	-	-	Minimal Embedding	P	LA	-
	VATNet	[49]	'19	E	Q	-	-	-	I3D [16], Faster R-CNN (RP only) [149]	P + I	FA	-
	VATT	[1]	'21	E	-	S	-	-	Minimal Embedding	P	LA	P
	MViT	[38]	'21	E	H	-	-	-	Minimal Embedding	P	LA	-
	SCT	[228]	'21	E	H	L	-	-	Minimal Embedding	P	LA	-
	CATE	[169]	'21	E	-	-	-	-	SlowFast [42] (Slow br.)	C	-	P
	LapFormer	[88]	'20	E	-	-	-	-	RN-50 [64]	P	FA	-
	TRX	[131]	'21	E	-	-	-	-	RN-50 [64]	F	FA	-
	LTT	[81]	'20	E	-	-	-	-	R(2+1)D [181]	F	LA	-
	Actor-T	[46]	'20	E	-	-	-	-	I3D [16], HRNet [184]	I	FA	-
	STiCA	[129]	'21	E	-	-	-	-	R(2+1)D-18 [181], RN-9 [64]	F	LA	A
	GroupFormer	[95]	'21	ED	Q	L	-	-	I3D [16]	I + F	LA	-
	Video Swin	[109]	'21	E	H	L	-	-	Minimal Embedding	P	LR	-
	VTN	[120]	'21	E	H	L	-	ViT [34]	Minimal Embedding	P	LA	-
	Video-Swin-V2	[107]	'22	E	H	L	-	-	RN-50 [64]	P	LR	P
	MTV	[211]	'22	E	H	-	-	ViT [34]	Minimal Embedding	P	LA	-
	Motionformer	[127]	'21	E	-	-	-	-	Minimal Embedding	P	LA	-
	X-ViT	[9]	'21	E	-	L	-	ViT [34]	Minimal Embedding	P	LA	-
	ObjTr	[200]	'21	E	-	-	-	-	Faster R-CNN [149], RN-101 [64]	I	FA + LA	-
	MViTv2	[97]	'22	E	H	-	-	-	Minimal Embedding	P	LR	-
	MaskFeat	[196]	'22	E	H	-	-	MViT [97]	Minimal Embedding	P	LR	P
	LSTCL	[185]	'22	E	-	-	-	Swin [109]	Minimal Embedding	P	LA	P
	RViT	[212]	'22	E	-	-	R	ViT [34]	Minimal Embedding	P	LA	-
	Direcformer	[182]	'22	E	-	A	-	TimeSformer [5]	Minimal Embedding	P	LA	-
	VideoMAE	[178]	'22	E	-	S*	-	ViT [34]	Minimal Embedding	P	LA	P
	BEVT	[187]	'22	E	H	L	-	Swin [109]	Minimal Embedding	P	LA	P
	TIME	[225]	'22	E	-	-	-	Motionformer [127]	Minimal Embedding	P	LA	A
	TokenLearner	[154]	'21	E	H	-	-	ViT [34]	Minimal Embedding	P	LA	-
	SVT	[146]	'22	E	-	A	-	TimeSformer [5]	Minimal Embedding	P	LA	P
	UniFormer	[93]	'22	E	H	L	-	-	Minimal Embedding	P	LA + LR *	-
Captioning	ActBERT	[239]	'20	E	-	-	-	-	R(2+1)D [181], Faster R-CNN [149]	I + C	LA	P
	HERO	[94]	'20	E	H	-	-	-	RN-101 [64], SlowFast [42]	F	FA	P
	MART	[92]	'20	ED	-	-	R	-	RN-200 [64], BN Inception [76]	F	FR	-
	VideoBERT	[168]	'19	E	-	-	-	-	S3D [205]	C	LA	P
	E2E-DC	[238]	'19	ED	-	-	-	-	RN-200 [64], BN Inception [76]	F	FA	-
	BMT	[74]	'20	ED	-	-	-	-	I3D [16]	F	FA	-
	AMT	[221]	'21	ED	-	-	-	-	RN-200 [64], BN-Inception [76]	F	FA	-
	MDVC	[75]	'20	ED	-	-	-	-	I3D [16]	F	FA	-
	RLM	[98]	'20	D	-	-	-	-	I3D [16]	C	FA	-
Retrieval	HiT	[104]	'21	E	-	-	-	-	S3D [205], SENet-154 [70]	F + C	LA	T
	COOT	[48]	'20	E	H	-	-	-	RN-152 [64]; ResNext-101 [204]; I3D [16]	F	-	T
	MMT	[45]	'20	E	-	-	-	-	S3D [205], DenseNet-101 [72], RN-50 [64], SENet-154 [70]	P + F	FA	T
	Support-set	[128]	'21	E	-	-	-	-	RN-152 [64], R(2+1)D-34	F	-	T
	TCA	[160]	'21	E	-	-	-	-	iMAC [53], L-3-iRMAC [89]	F	-	T
	MDMMT	[36]	'21	E	-	-	-	-	CLIP [140]	F	LA	T
	Fast and Slow	[117]	'21	D	-	-	-	-	TSM RN-50 [100]	P	-	T
	ClipBERT	[91]	'21	E	-	S*	-	-	RN-50 [64]	P	LA	-
	CACL	[58]	'22	E	-	-	-	-	RN-50 [64]	F	LA	P
Tracking	Hopper	[237]	'21	ED	-	-	-	-	ResNeXt-101 [204], DETR [12]	I + F	LA	-
	DTT	[218]	'21	ED	-	-	-	-	RN-50 [64]	P	LA	-
	TrDIMP	[186]	'21	ED	-	-	-	-	RN-50 [64]	P	-	-
	TransT	[23]	'21	E	-	-	-	-	RN-50 [64]	P	FA	-
	STARK	[210]	'21	ED	-	-	-	-	RN-50 [64]	P	FA	-
	Trackformer	[116]	'22	ED	Q	-	MR	-	RN-50 [64]	P	FA	-
	VDRFormer	[236]	'22	ED	Q	-	MR	-	RN-101 [64]	P	FA	-

*: Non-attentional sparsity (e.g., input level).

Table C.3: General overview of relevant Video Transformers surveyed. In *Architecture*, "Arch.": architecture, that is Encoder (E), Decoder (D), or Encoder-Decoder (ED); "Aggr.", aggregation strategy, either Hierarchical (H) or Query-driven compression (Q); "Restriction", can be Local (L), Axial (A), Sparse (S), or a mix. "Long-t.": long-term temporal modeling, Memory (M), Recurrence (R), or a both. In *Input*, "Backbone" refers to Transformer backbone; "Embd.", the Embedding Network; "Tknz', the tokenization strategy, patch- (P), instance- (I), frame- (F), or clip-wise (C); and "Pos.", the positional embedding, can be Fixed Absolute (FA), Fixed Relative (FR), Learned Absolute (LA), Learned Relative (LR), or a combination. (Continuation in table C.4)

is enlarged across stages by subsequently merging token embeddings. Others pursue building very deep Transformers by maintaining a very compact latent representation [77]. These larger Transformers for classification require

	Name	Ref.	Yr.	Architecture				Input				Train.
				Arch.	Aggr.	Restr.	Long-t.	Backbone	Embd.	Tknz.	Pos.	SSL
Low-level	ET-Net	[198]	'21	ED	-	-	-	-	ConvLSTM [163]	P	FA	T
	STTN	[227]	'20	ED	-	-	-	-	2D CNN (custom)	P	-	T
	FuseFormer	[103]	'21	ED	-	-	-	-	I3D [16]	P	-	T
	SAVM	[197]	'20	ED	-	L	-	-	Minimal Embeddings	P	LR	T
	VLT	[143]	'20	ED	-	-	-	-	VQ-VAE [122]	P	FR	T
	TransformerFusion	[7]	'21	E	-	S*	M	-	RN-18 [64]	F	LA	-
Segmentation	VisTR	[194]	'21	ED	-	-	-	-	RN-50 [64]	P	FA	-
	MFN	[193]	'21	E	-	-	-	-	3D CNN (custom)	P	FA	-
	CMSANet	[216]	'21	E	-	-	-	-	DeepLab-101 [19]	P	FA	-
	IFC	[73]	'22	ED	Q	-	-	-	RN-101 [64]	P	FA	-
	TeViT	[213]	'22	ED	Q	-	-	MsgShifT [188, 213]	Minimal Embedding	P	FA	-
	AOT	[215]	'21	E	-	L	MR	Swin [109]	MobileNet-V2 [155]	P	FA + RL	-
O.D.	PCSA	[57]	'20	E	-	L	-	-	MobileNet-V3 [69]	P	-	-
	TCTR	[224]	'21	ED	-	-	-	-	RN-50 [64]	P	FA	-
	PMPNet	[217]	'20	ED	-	-	-	-	GraphCNN (custom)	P	-	-
	ORViT	[68]	'22	ED	-	-	-	-	Faster R-CNN [149], RN-50 [64]	P + I	LR	-
Summ.	H-MAN	[106]	'19	E	-	-	-	-	VAE-GAN [112]	F	-	-
	VasNet	[37]	'19	E	-	-	-	-	GoogLeNet [173]	F	FA	-
	BiDAVS	[101]	'20	E	-	-	-	-	GoogLeNet [173]	F	LR	-
	VMTN	[157]	'19	E	Q	-	-	-	ResNet-18 [64], SENet-101 [70]	P	FA	-
Localiz.	HISAN	[134]	'19	E	-	-	-	-	Faster R-CNN [149]	I + F	-	-
	STVGBert	[166]	'21	E	Q	-	-	-	RN-101 [64]	P	-	-
	MeMViT	[201]	'22	E	H	-	M	MViTv2 [97]	Minimal Embedding	P	LR	-
	MSAT	[232]	'21	E	-	-	-	-	C3D [179]	C	FA	-
	RTD-Net	[174]	'21	D	-	-	-	-	I3D [16]	F	LR	-
	LSTR	[206]	'21	ED	Q	-	M	-	RN-50 [64]	F	FA	-
Others	SiaSamRea	[220]	'21	E	-	S*	-	ClipBERT [91]	RN-50 [64]	P	LA	A
	Perceiver	[77]	'21	E	Q	-	-	-	Minimal Embedding	P	LA	-
	AVT	[50]	'21	E	H	-	-	ViT [34]	Minimal Embedding	P	LA	A
	OadTR	[189]	'21	ED	-	-	-	-	RN-200 [64], BN-Inception [76]	F	LA	-
	STTran	[27]	'21	ED	-	L	-	-	RN-101 F R-CNN [149]	I + F	LA	-
	E.T.	[125]	'21	E	-	-	-	-	Faster R-CNN [149], Mask R-CNN [63]	F	FA	-
	SMT	[39]	'19	ED	Q	S*	M	-	RN-18 [64]	F	FA	-
	JSLT	[11]	'20	ED	-	-	-	-	InceptionV4 [172]	F	FA	-
	MSLT	[10]	'20	ED	-	-	-	-	InceptionV4 [172]	F	FA	-
	SBL	[111]	'20	ED	-	-	-	-	RN-18 [64]	F	-	-
	MDAM	[86]	'19	E	Q	-	-	-	RN-152 [64]	F	FA	-
	PSAC	[234]	'21	E	-	-	-	-	Minimal Embedding	P	FA	-
	BTH	[96]	'21	E	-	-	-	-	VGG-16 [165]	F	FA	P
	BERT4SessRec	[25]	'20	E	-	-	-	-	GoogLeNet [173]	C	FA	P
	Dyadformer	[28]	'21	E	-	-	-	-	R(2+1)D-152 [181]	C	FA	-
	MM-Transformer	[152]	'22	ED	-	L	-	-	Mask R-CNN [63]	I	FA	-

*: Non-attentional sparsity (e.g., input level)

Table C.4: (Continuation of table C.3)

large labeled datasets for fully-supervised training [38, 109] or heavily rely on self-supervised pre-training [90, 168]. For multi-modal datasets, encoder fusion [168] or hierarchical encoder fusion is utilized [167].

Several other works rely on larger (usually CNN-based) backbones [46, 81, 90, 129, 131, 167–169], facilitating the training on smaller datasets. When equipped with these backbones, shallow encoders can serve as mere pooling operators [46, 81, 129, 137]. For detection backbones, Transformers are also a natural way to fuse information among detections [95] or to allow them to attend over a larger visual context [49]. Although mostly used in pure Transformers, efficient designs have been explored for these kinds of works as well, e.g. weight sharing [90].

8.2 Video translation

The translation task intends to map the raw input video to an output signal of a potentially different nature and with an arbitrary (a priori unknown) length. Although the output could be another video, it is often a signal in another modality (e.g., language) or simply a sequence of discrete symbols. The most popular instantiation of translation is *video captioning* [74, 92, 94, 117] that consists in producing natural language descriptions of what is globally going on in the video. When producing separate captions for different video subparts independently, this is referred to as *dense video captioning* [221, 238]. A more specialized type of video captioning is *sign-language translation* [10, 11]. Additional other forms of translation are: *video reasoning* [237], which extends the task of captioning by allowing a natural language prompt along with the video; *video-language dialogue* systems, which add to reasoning the requirement of back and forth communication with an external agent while reasoning about the visuals [98]; *temporal (or spatiotemporal) action localization* [174] to produce a list of, respectively, temporal begin and end times or a "tube" of bounding boxes containing the human actions in the video; or *robot video-based navigation* [39], in which the video – and perhaps other sensory inputs – are translated to the next action (a sequence of next actions) to take.

VTs tackling translation typically leverage encoder-decoder architectures, in which video is passed through the encoder and served as context to the decoder – similarly to [183], only that the encoder is a video encoder instead of a language one. Task-specific modifications of this design are found for dense video captioning [74, 221, 238], where a temporal proposal generator is attached after the encoder to tell the decoder where/when in the sequence it has to focus. [74] is a two-stage method where the proposals are generated in the first stage. In the second stage, proposals are used to cut temporal clips from the video that need to be re-encoded (to avoid information from the different proposals being mixed up) and fed to the decoder to produce the per-clip captions. Slightly different is [238], which instead of clipping the videos, converts the proposals to differentiable masks with a masking function whose parameters are trained also getting a back-propagation signal from the decoder. Still, the different masks have to be applied to the video and yet again re-forwarded through the encoder. [221] eliminates the re-forwarding by making the most of local self-attention, which limits the leak of information across the encoded proposals. [174] tackled temporal action detection by relying only on a Transformer Decoder. Inspired by [12], proposals are not generated by the encoder or an external module after it but are sourced from a set of learnable token embeddings input to the decoder. The decoder augments these tokens and, later, two heads are in charge of classifying those into actions and regressing their temporal position and length – similarly to a YOLO-like network [148].

In most of those works, the decoder module maintained its canonical form, although there are works that propose small variations. One has to do with the first and foremost SA sublayer. [98] removes the decoder's SA layer before the CA, whereas [221] substitutes it by a moving average – both to make the models computationally lighter. Another one is to modify the input received by the CA sublayer. [74, 238] receive the outputs of the encoder layers but at their respective depth, instead of only the one from the encoder's layer. The disadvantage of this is assuming the encoder and the decoder require the same number of layers. In [10], the decoder also receives multiple inputs but from separate encoders. The decoder deals with those in different parallel CA sublayers and averages their outputs. There are also designs that go without a Transformer encoder, replacing it with an external non-Transformer module [174] or relying entirely on the decoder [92, 98]. [98] follow the prompt-based input of GPT-2 [142] and feds n video features as the first tokens in the decoding sequence and decodes the caption starting at the $n + 1$-th input embedding. In particular, [92] prompts not only the visual features but the current language sentence features to generate the next sentence in a paragraph. All in all, prompting is the generalization of the original shifting operation in [183], where the decoding starts at a shifted position to account for the start token.

8.3 Video retrieval

The task of retrieval consists in recovering a piece of information associated to a particular query. Those associations can be video-video pairs [160] or pairs composed of different modalities (video with, most often, language [48, 104, 117, 128] or language plus audio [36, 45]). Retrieval relies on a distance metric among the representations of the queries and the retrieval candidates. The representations are learned during training using the pairs to minimize the distances between the representations of the corresponding pairs while repelling from the query the non-corresponding candidates' representations in a joint space. This can be done through classification, by extending BERT's *Next Sentence Prediction* to a cross-modal matching task, forcing the network to find co-occurrent information in both modalities [90, 168, 239]. Alternatively, this can be naturally extended into a contrastive setting. In retrieval, it is common to use two anchors (which form the positive pair) and two negative sets, one from each modality. In VT literature we find these losses instantiated through a combined hinge loss [48, 94] or *Bi-directional Max-Margin* [36, 45, 128], which enforce similarity for true pairs to be higher than that of negative pairs, by at least a given margin. Alternatively, InfoNCE is also used [104, 129], normalizing the similarity score of positive pairs by that of a set of negative pairs, effectively forcing the network to learn similar representations for correctly paired samples and vice-versa for negative ones. While the most common approach is to align final output representations, some works leverage hier-

archical contrastive losses, which also align intermediate feature representations [48, 104]. During inference, the aligned representations are fixed, so the task simply becomes a search (e.g., K-Nearest Neighbors) to find the top-k examples most similar to a given query within the database of candidates' pre-computed representations.

One interesting variation of this pipeline is [117], in which the alignment is performed on the outputs of a siamese two-stream video-and-language CNN for faster retrieval instead. Then, a decoder-only Transformer fed with the text as input and CNN-based video features as context re-ranks the previously top-k retrieved elements using the decoding likelihood score. In a similar spirit, [128] also leverages the likelihood of a language-based decoder-only Transformer during training as a loss that measures how well the query language caption can be reconstructed from the weighted combination of the features from all the non-corresponding videos in the batch. Those weights are based on the similarity of the query caption with the captions of those other videos. [45] aligns at the same time video, audio, and recognized speech with a language caption. The language-video, language-audio, and language-speech similarities are aggregated before contrastive alignment with a mixture of weights governed by the content of the caption (e.g., the language-video similarity is given more weight if the caption refers to something that is more salient in the video than in the other modalities).

8.4 Object-centric tasks: tracking and object detection

Tasks such as object detection, tracking, and segmentation are inherently object-centric in nature, and recent work [68, 73, 116, 213, 236] within these tasks have begun to leverage temporally coherent object representations. As object-centric approaches tend to focus on per-object outputs, a large part of the information within a given frame is redundant (as mentioned in Sec. 4.1), therefore leveraging known and relevant content from previous frames can be used to focus the global attention to object relevant cues. As such, these approaches typically leverage memory or recurrency (as described in 4.2) to correlate object information temporally.

In the former recent work [73, 213] leverage a set of "messenger tokens to relay contextual information between frames. IFC-transformer [73] processes the relationship in an isolated encoder, whereas TeViT [213] shifts the tokens between frame sequences to achieve object-specific information aggregation, to accumulate temporal information across different steps sequentially, resulting in a hierarchical-like approach for temporal information sharing. Other work [215] however, performs both long-term and short-term information sharing in parallel, subsequently concatenated. Due to varied framerate and inter-frame changes in content, smoothness cannot be guaranteed through long-term alone, thus short-term attention is computed on a smaller

spatiotemporal neighborhood, to ensure smooth and continuous predictions between frames. With regards to recurrency, other approaches [116, 236] leverage object-specific tokens that are derived from an outputted bounding box, and the spatial+size information is then recurrently propogated [116, 236] or used to produce region-specific attention for each concurrently between frames [68]. Inspired by recent works in Vision transformers (particularly DETR and its variants) [116, 236], leverages the bounding box predictions from each frame to augment the decoder queries by concatenating detection tokens from previous frames to the existing learned-fixed tokens, in addition to storing each detection in memory for increased robustness to occlusions in the video sequence.

As can be observed in vision transformers, architectures that leverage the object features to aggregate contextual information such as [68, 134] attempt to enhance existing representations with more focus on object-centric information. Where ORViT [68] leverages auxiliary bounding box information in each transformer layer, whereas the GroupFormer [95] leverages bounding boxes to isolate objects for a separate object-specific action classification branch. Unlike the recurrent and memory style approach these types of approaches don't seem to aim for an efficient design in terms of computational cost, but rather efficient in the sense of information-rich representation, that leverages object-centric information in addition to global context information.

8.5 Low-level tasks

Given the high dimensionality of video data, video generation tasks are quite challenging, and not many video Transformers try to address them. In particular, [143, 197] tackle future frame prediction, [198] generates grayscale video from event-based videos and [103, 227] perform video inpainting. Most of these propose to embed a Transformer within some type of convolutional auto-encoder to evolve representations between encoder and decoder [103, 198, 227]. The only exception is [197], which performs local attention and generates video autoregressively one pixel channel at a time. Interestingly, [103] outperforms [227] in all tested benchmarks for inpainting by using an overlapping patch tokenization strategy.

8.6 Segmentation

Most work in segmentation leverage temporal relations to refine intermediate feature representations [193, 194, 219]. Most notably, [194] leverages the Transformers' ability to view the entire sequence, to include an auxiliary loss where representations of individuals are matched temporally, effectively teaching the network to implicitly track objects and leverage temporal fine-grained information. Alternatively, [216] leverages a novel word-visual attention mechanism

allowing a textual query to attend to specific content in multiple spatial scales and perform segmentation based on the said query.

8.7 Summarization

Few works have used Transformers for the task of video summarization by predicting frame-wise importance scores. We find two key trends when solving this task through VTs: the use of RNNs as an initial step [106, 171] and using individual frames to attend to aggregated subsets of the video either from a GRU [171] or by using a masked Transformer [101].

8.8 Other tasks

Transformers have also been applied for action anticipation [50, 189], sign-language translation [10, 11], visual-question answering [86, 234], autonomous driving [133], robot navigation [39], visual-language navigation [125], personality recognition [28], lip reading [111], dynamic scene graph generation [27], and multimedia recomendation [25]. As not many video Transformers have tackled this, it is too early to ascertain specific trends, so we simply list them here for completeness.

References

[1] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," *NeurIPS*, 2021.

[2] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021.

[3] A. Bardes, J. Ponce, and Y. Lecun, "Vicreg: Variance-invariance-covariance regularization for self-supervised learning," in *ICLR*, 2022.

[4] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv*, 2020.

[5] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, 2021.

[6] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *ICCV*, 2021.

[7] A. Bozic, P. Palafox, J. Thies, A. Dai, and M. Nießner, "Transformerfusion: Monocular rgb scene reconstruction using transformers," *NeurIPS*, 2021.

[8] S. Buch, C. Eyzaguirre, A. Gaidon, J. Wu, L. Fei-Fei, and J. C. Niebles, "Revisiting the "video" in video-language understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[9] A. Bulat, J. M. Perez Rua, S. Sudhakaran, B. Martinez, and G. Tzimiropoulos, "Space-time mixing attention for video transformer," *NeurIPS*, 2021.

[10] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Multi-channel transformers for multi-articulatory sign language translation," in *ECCV*, 2020.

[11] ——, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *CVPR*, 2020.

[12] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.

[13] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *ICCV*, 2021.

[14] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," *arXiv*, 2019.

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[16] ——, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.

[17] C.-F. R. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan, "Deep analysis of cnn-based spatio-temporal representations for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6165–6175.

[18] J. Chen and H. Chao, "Videotrm: Pre-training for video captioning challenge 2020," in *ACM-MM*, 2020.

[19] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *TPAMI*, 2017.

[20] T. Chen, Z. Zhang, Y. Cheng, A. Awadallah, and Z. Wang, "The principle of diversity: Training stronger vision transformers calls for reducing all levels of redundancy," in *CVPR*, 2022.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020.

[22] X. Chen, C.-J. Hsieh, and B. Gong, "When vision transformers outperform resnets without pretraining or strong data augmentations," *arXiv*, 2021.

[23] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *CVPR*, 2021.

[24] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021.

[25] X. Chen, D. Liu, C. Lei, R. Li, Z.-J. Zha, and Z. Xiong, "Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer," in *ACM-MM*, 2019.

[26] Y. Chen and J. Joo, "Understanding and mitigating annotation bias in facial expression recognition," in *ICCV*, 2021.

[27] Y. Cong, W. Liao, H. Ackermann, B. Rosenhahn, and M. Y. Yang, "Spatial-temporal transformer for dynamic scene graph generation," in *ICCV*, 2021.

[28] D. Curto, A. Clapes, J. Selva, S. Smeureanu, J. C. S. J. Junior, D. Gallardo-Pujol, G. Guilera, D. Leiva, T. B. Moeslund, S. Escalera, and C. Palmero, "Dyadformer: A multi-modal transformer for long-range modeling of dyadic interactions," in *ICCV-W*, 2021.

[29] Z. Dai, H. Liu, Q. V. Le, and M. Tan, "Coatnet: Marrying convolution and attention for all data sizes," in *NeurIPS*, 2021.

[30] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Computational Linguistics*, 2019.

[33] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," in *International Conference on Machine Learning*. PMLR, 2021, pp. 2793–2803.

[34] 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," in *ICLR*, 2021.

[35] B. Duke, A. Ahmed, C. Wolf, P. Aarabi, and G. W. Taylor, "Sstvos: Sparse spatiotemporal transformers for video object segmentation," in *CVPR*, 2021.

[36] M. Dzabraev, M. Kalashnikov, S. Komkov, and A. Petiushko, "Mdmmt: Multidomain multimodal transformer for video retrieval," in *CVPR*, 2021.

[37] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, "Summarizing videos with attention," in *ACCV*, 2018.

[38] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *ICCV*, 2021.

[39] K. Fang, A. Toshev, L. Fei-Fei, and S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," in *CVPR*, 2019.

[40] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *CVPR*, 2020.

[41] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *ICCV*, 2019.

[42] ——, "Slowfast networks for video recognition," in *ICCV*, 2019.

[43] C. Feichtenhofer, H. Fan, B. Xiong, R. Girshick, and K. He, "A large-scale study on unsupervised spatiotemporal representation learning," in *CVPR*, 2021.

[44] Q. Fournier, G. M. Caron, and D. Aloise, "A practical survey on faster and lighter transformers," *arXiv*, 2021.

[45] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal Transformer for Video Retrieval," in *ECCV*, 2020.

[46] K. Gavrilyuk, R. Sanford, M. Javan, and C. G. Snoek, "Actor-transformers for group activity recognition," in *CVPR*, 2020.

[47] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *CVPR*, 2019.

[48] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox, "Coot: Cooperative hierarchical transformer for video-text representation learning," in *NeurIPS*, 2020.

[49] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *CVPR*, 2019.

[50] R. Girdhar and K. Grauman, "Anticipative video transformer," in *ICCV*, 2021.

[51] R. Girdhar, M. Singh, N. Ravi, L. van der Maaten, A. Joulin, and I. Misra, "Omnivore: A single model for many visual modalities," in *CVPR*, 2022.

[52] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Interspeech*, 2021.

[53] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *ICCV*, 2017.

[54] D. Gordon, K. Ehsani, D. Fox, and A. Farhadi, "Watching the world go by: Representation learning from unlabeled videos," *arXiv*, 2020.

[55] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The "something something" video database for learning and evaluating visual common sense," in *ICCV*, 2017.

[56] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *NeurIPS*, 2020.

[57] Y. Gu, L. Wang, Z. Wang, Y. Liu, M.-M. Cheng, and S.-P. Lu, "Pyramid constrained self-attention network for fast video salient object detection," in *AAAI*, 2020.

[58] S. Guo, Z. Xiong, Y. Zhong, L. Wang, X. Guo, B. Han, and W. Huang, "Cross-architecture self-supervised video representation learning," in *CVPR*, 2022.

[59] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," in *IEEE TPAMI*, 2022.

[60] T. Han, W. Xie, and A. Zisserman, "Video representation learning by dense predictive coding," in *ICCV-W*, 2019.

[61] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *CVPR*, 2022.

[62] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020.

[63] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[65] D. Hendrycks, X. Liu, E. Wallace, A. Dziedzic, R. Krishnan, and D. Song, "Pretrained transformers improve out-of-distribution robustness," in *ACL*, 2020.

[66] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," *NeurIPS*, 2019.

[67] Y.-J. Heo, Y.-J. Choi, Y.-W. Lee, and B.-G. Kim, "Deepfake detection scheme based on vision transformer and distillation," *arXiv*, 2021.

[68] R. Herzig, E. Ben-Avraham, K. Mangalam, A. Bar, G. Chechik, A. Rohrbach, T. Darrell, and A. Globerson, "Object-region video transformers," in *CVPR*, 2022.

[69] A. Howard, M. Sandler, G. Chu, and L.-C. e. a. Chen, "Searching for mobilenetv3," in *CVPR*, 2019.

[70] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018.

[71] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *CVPR*, 2018.

[72] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[73] S. Hwang, M. Heo, S. W. Oh, and S. J. Kim, "Video instance segmentation using inter-frame communication transformers," *NeurIPS*, 2021.

[74] V. Iashin and E. Rahtu, "A better use of audio-visual cues: Dense video captioning with bi-modal transformer," in *BMVC*, 2020.

[75] ——, "Multi-modal dense video captioning," in *CVPR*, 2020.

[76] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.

[77] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," in *ICML*, 2021.

[78] S. Jain and B. C. Wallace, "Attention is not explanation," in *NAACL-HLT*, 2019.

[79] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE TPAMI*, 2020.

[80] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *CVPR*, 2020.

[81] M. E. Kalfaoglu, S. Kalkan, and A. A. Alatan, "Late temporal modeling in 3d cnn architectures with bert for action recognition," in *ECCV*, 2020.

[82] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "AMMUS : A survey of transformer-based pretrained models in natural language processing," *arXiv*, 2021.

[83] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM CSUR*, 2022.

[84] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, "Selfreg: Self-supervised contrastive regularization for domain generalization," in *ICCV*, 2021.

[85] D. Kim, D. Cho, and I. S. Kweon, "Self-supervised video representation learning with space-time cubic puzzles," in *AAAI*, 2019.

[86] K.-M. Kim, S.-H. Choi, J.-H. Kim, and B.-T. Zhang, "Multimodal dual attention memory for video story question answering," in *ECCV*, 2018.

[87] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *ICLR*, 2019.

[88] S. Kondo, "Lapformer: surgical tool detection in laparoscopic surgical video using transformer architecture," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2020.

[89] G. Kordopatis-Zilos, S. Papadopoulos, I. Patras, and I. Kompatsiaris, "Visil: Fine-grained spatio-temporal video similarity learning," in *ICCV*, 2019.

[90] S. Lee, Y. Yu, G. Kim, T. Breuel, J. Kautz, and Y. Song, "Parameter efficient multimodal transformers for video representation learning," in *ICLR*, 2021.

[91] J. Lei, L. Li, L. Zhou, Z. Gan, T. L. Berg, M. Bansal, and J. Liu, "Less is more: Clipbert for video-and-language learning via sparse sampling," in *CVPR*, 2021.

[92] J. Lei, L. Wang, Y. Shen, D. Yu, T. L. Berg, and M. Bansal, "Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning," in *ACL*, 2020.

[93] K. Li, Y. Wang, G. Peng, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unified transformer for efficient spatial-temporal representation learning," in *ICLR*, 2022.

[94] L. Li, Y.-C. Chen, Y. Cheng, Z. Gan, L. Yu, and J. Liu, "Hero: Hierarchical encoder for video+ language omni-representation pre-training," in *EMNLP*, 2020.

[95] S. Li, Q. Cao, L. Liu, K. Yang, S. Liu, J. Hou, and S. Yi, "Groupformer: Group activity recognition with clustered spatial-temporal transformer," in *ICCV*, 2021.

[96] S. Li, X. Li, J. Lu, and J. Zhou, "Self-supervised video hashing via bidirectional transformers," in *CVPR*, 2021.

[97] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *CVPR*, 2022.

[98] Z. Li, Z. Li, J. Zhang, Y. Feng, and J. Zhou, "Bridging text and video: A universal multimodal transformer for audio-visual scene-aware dialog," *IEEE/ACM TASLP*, 2021.

[99] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *ICCV*, 2019.

[100] ——, "Tsm: Temporal shift module for efficient video understanding," in *ICCV*, 2019.

[101] J. Lin and S.-h. Zhong, "Bi-directional self-attention with relative positional encoding for video summarization," in *ICTAI*, 2020.

[102] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[103] R. Liu, H. Deng, Y. Huang, X. Shi, L. Lu, W. Sun, X. Wang, J. Dai, and H. Li, "Fuseformer: Fusing fine-grained information in transformers for video inpainting," in *ICCV*, 2021.

[104] S. Liu, H. Fan, S. Qian, Y. Chen, W. Ding, and Z. Wang, "Hit: Hierarchical transformer with momentum contrast for video-text retrieval," in *ICCV*, 2021.

[105] Y. Liu, Y. Zhang, Y. Wang, F. Hou, J. Yuan, J. Tian, Y. Zhang, Z. Shi, J. Fan, and Z. He, "A Survey of Visual Transformers," *arXiv*, 2021.

[106] Y.-T. Liu, Y.-J. Li, F.-E. Yang, S.-F. Chen, and Y.-C. F. Wang, "Learning hierarchical self-attention for video summarization," in *ICIP*, 2019.

[107] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *CVPR*, 2022.

[108] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[109] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *CVPR*, 2022.

[110] K. Lu, A. Grover, P. Abbeel, and I. Mordatch, "Frozen pretrained transformers as universal computation engines," *AAAI CAI*, 2022.

[111] M. Luo, S. Yang, X. Chen, Z. Liu, and S. Shan, "Synchronous bidirectional learning for multilingual lip reading," in *BMVC*, 2020.

[112] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial lstm networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 202–211.

[113] F. Mahdisoltani, G. Berger, W. Gharbieh, D. Fleet, and R. Memisevic, "On the effectiveness of task granularity for transfer learning," *arXiv*, 2018.

[114] K. Mahmood, R. Mahmood, and M. Van Dijk, "On the robustness of vision transformers to adversarial examples," in *ICCV*, 2021.

[115] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*, 1989.

[116] T. Meinhardt, A. Kirillov, L. Leal-Taixe, and C. Feichtenhofer, "Trackformer: Multi-object tracking with transformers," in *CVPR*, 2022.

[117] A. Miech, J.-B. Alayrac, I. Laptev, J. Sivic, and A. Zisserman, "Thinking fast and slow: Efficient text-to-visual retrieval with transformers," in *CVPR*, 2021.

[118] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *ICCV*, 2019.

[119] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, "Intriguing properties of vision transformers," *NeurIPS*, 2021.

[120] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *ICCV*, 2021.

[121] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv*, 2018.

[122] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," *arXiv*, 2017.

[123] S. Oprea, P. Martinez-Gonzalez, A. Garcia-Garcia, J. A. Castro-Vargas, S. Orts-Escolano, J. Garcia-Rodriguez, and A. Argyros, "A review on deep learning techniques for video prediction," *IEEE TPAMI*, 2020.

[124] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Empirical Methods in Natural Language Processing*, 2016.

[125] A. Pashevich, C. Schmid, and C. Sun, "Episodic transformer for vision-and-language navigation," in *ICCV*, 2021.

[126] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.

[127] M. Patrick, D. Campbell, Y. Asano, I. Misra, F. Metze, C. Feichtenhofer, A. Vedaldi, and J. F. Henriques, "Keeping your eye on the ball: Trajectory attention in video transformers," *NeurIPS*, 2021.

[128] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. G. Hauptmann, J. F. Henriques, and A. Vedaldi, "Support-set bottlenecks for video-text representation learning," in *ICLR*, 2021.

[129] M. Patrick, P.-Y. Huang, I. Misra, F. Metze, A. Vedaldi, Y. M. Asano, and J. a. F. Henriques, "Space-time crop & attend: Improving cross-modal video representation learning," in *ICCV*, 2021.

[130] S. Paul and P.-Y. Chen, "Vision transformers are robust learners," in *AAI CAI*, 2022.

[131] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, and D. Damen, "Temporal-relational crosstransformers for few-shot action recognition," in *CVPR*, 2021.

[132] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *CVIU*, 2021.

[133] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *CVPR*, 2021.

[134] R. R. A. Pramono, Y.-T. Chen, and W.-H. Fang, "Hierarchical self-attention network for action localization in videos," in *CVPR*, 2019.

[135] S. Purushwalkam and A. Gupta, "Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases," *NeurIPS*, 2020.

[136] D. Purwanto, R. R. A. Pramono, Y.-T. Chen, and W.-H. Fang, "Three-stream network with bidirectional self-attention for action recognition in extreme low resolution videos," *IEEE Signal Processing Letters*, 2019.

[137] D. Purwanto, R. Renanda Adhi Pramono, Y.-T. Chen, and W.-H. Fang, "Extreme low resolution action recognition with spatial-temporal multi-head self-attention and knowledge distillation," in *CVPR*, 2019.

[138] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *NeurIPS*, 2017.

[139] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *CVPR*, 2021.

[140] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," *arXiv*, 2021.

[141] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," in *OpenAI Preprint*, 2018.

[142] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," in *OpenAI Blog*, 2019.

[143] R. Rakhimov, D. Volkhonskiy, A. Artemov, D. Zorin, and E. Burnaev, "Latent video transformer," *arXiv*, 2020.

[144] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *NeurIPS*, 2019.

[145] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *ICML*, 2021.

[146] K. Ranasinghe, M. Naseer, S. Khan, F. S. Khan, and M. S. Ryoo, "Self-supervised video transformer," in *CVPR*, 2022.

[147] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, 1999.

[148] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[149] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015.

[150] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "Imagenet-21k pretraining for the masses," in *NeurIPS*, 2021.

[151] F. Rodrigues and F. Pereira, "Deep learning from crowds," in *AAAI Conference on AI*, 2018.

[152] D. Roy and B. Fernando, "Action anticipation using pairwise human-object interactions and transformers," *IEEE TIP*, 2021.

[153] L. Ruan and Q. Jin, "Survey: Transformer based video-language pre-training," *AI Open*, 2022.

[154] M. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: Adaptive space-time tokenization for videos," *NeurIPS*, 2021.

[155] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018.

[156] M. C. Schiappa, Y. S. Rawat, and M. Shah, "Self-supervised learning for videos: A survey," *arXiv*, 2022.

[157] H. Seong, J. Hyun, and E. Kim, "Video multitask transformer network," in *ICCV*, 2019.

[158] S. Serrano and N. A. Smith, "Is attention interpretable?" in *ACL*, 2019.

[159] L. Sevilla-Lara, S. Zha, Z. Yan, V. Goswami, M. Feiszli, and L. Torresani, "Only time can tell: Discovering temporal data for temporal modeling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[160] J. Shao, X. Wen, B. Zhao, and X. Xue, "Temporal context aggregation for video retrieval with contrastive learning," in *WACV*, 2021.

[161] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," in *NAACL*, 2018.

[162] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NeurIPS*, 2015.

[163] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[164] A. Shin, M. Ishii, and T. Narihira, "Perspectives and prospects on transformer architecture for cross-modal tasks with language and vision," *IJCV*, 2022.

[165] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.

[166] R. Su, Q. Yu, and D. Xu, "Stvgbert: A visual-linguistic transformer based framework for spatio-temporal video grounding," in *ICCV*, 2021.

[167] C. Sun, F. Baradel, K. Murphy, and C. Schmid, "Contrastive bidirectional transformer for temporal representation learning," *arXiv*, 2019.

[168] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *ICCV*, 2019.

[169] C. Sun, A. Nagrani, Y. Tian, and C. Schmid, "Composable augmentation encoding for video representation learning," in *ICCV*, 2021.

[170] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," in *CVPR*, 2022.

[171] Y.-L. Sung, C.-Y. Hong, Y.-C. Hsu, and T.-L. Liu, "Video summarization with anchors and multi-head attention," in *ICIP*, 2020.

[172] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *AAAI*, 2017.

[173] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[174] J. Tan, J. Tang, L. Wang, and G. Wu, "Relaxed transformer decoders for direct action proposal generation," in *ICCV*, 2021.

[175] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM CSUR*, 2020.

[176] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," in *CVPR*, 2022.

[177] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *ICML*, 2021.

[178] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *NeurIPS*, 2022.

[179] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[180] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *CVPR*, 2019.

[181] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *CVPR*, 2018.

[182] T.-D. Truong, Q.-H. Bui, C. N. Duong, H.-S. Seo, S. L. Phung, X. Li, and K. Luu, "Direcformer: A directed attention in transformer approach to robust action recognition," in *CVPR*, 2022.

[183] 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 *NeurIPS*, 2017.

[184] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *TPAMI*, 2020.

[185] J. Wang, G. Bertasius, D. Tran, and L. Torresani, "Long-short temporal contrastive learning of video transformers," in *CVPR*, 2022.

[186] N. Wang, W. Zhou, J. Wang, and H. Li, "Transformer meets tracker: Exploiting temporal context for robust visual tracking," in *CVPR*, 2021.

[187] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, Y.-G. Jiang, L. Zhou, and L. Yuan, "Bevt: Bert pretraining of video transformers," in *CVPR*, 2022.

[188] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, 2022.

[189] X. Wang, S. Zhang, Z. Qing, Y. Shao, Z. Zuo, C. Gao, and N. Sang, "Oadtr: Online action detection with transformers," in *ICCV*, 2021.

[190] X. Wang, X. Xiong, M. Neumann, A. Piergiovanni, M. S. Ryoo, A. Angelova, K. M. Kitani, and W. Hua, "Attentionnas: Spatiotemporal attention cell search for video classification," in *ECCV*, 2020.

[191] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *CVPR*, 2018.

[192] X. Wang and A. Gupta, "Unsupervised learning of visual representations using videos," in *ICCV*, 2015.

[193] Y. Wang, Z. Liu, Y. Xia, C. Zhu, and D. Zhao, "Spatiotemporal module for video saliency prediction based on self-attention," *Image and Vision Computing*, 2021.

[194] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, "End-to-end video instance segmentation with transformers," in *CVPR*, 2021.

[195] ——, "End-to-end video instance segmentation with transformers," in *CVPR*, 2021.

[196] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *CVPR*, 2022.

[197] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," in *ICLR*, 2020.

[198] W. Weng, Y. Zhang, and Z. Xiong, "Event-based video reconstruction using transformer," in *ICCV*, 2021.

[199] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," in *EMNLP-IJCNLP*, 2019.

[200] C.-Y. Wu and P. Krahenbuhl, "Towards long-form video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1884–1894.

[201] C.-Y. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer, "Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition," in *CVPR*, 2022.

[202] K. Wu, H. Peng, M. Chen, J. Fu, and H. Chao, "Rethinking and improving relative position encoding for vision transformer," in *ICCV*, 2021.

[203] F. Xiao, K. Kundu, J. Tighe, and D. Modolo, "Hierarchical self-supervised representation learning for movie understanding," in *CVPR*, 2022.

[204] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *CVPR*, 2017.

[205] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *ECCV*, 2017.

[206] M. Xu, Y. Xiong, H. Chen, X. Li, W. Xia, Z. Tu, and S. Soatto, "Long short-term transformer for online action detection," *NeurIPS*, 2021.

[207] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: a survey," *arXiv*, 2022.

[208] Q. Xu, M. Zhang, Z. Gu, and G. Pan, "Overfitting remedy by sparsifying regularization on fully-connected layers of cnns," *Neurocomputing*, 2019.

[209] Y. Xu, H. Wei, M. Lin, Y. Deng, K. Sheng, M. Zhang, F. Tang, W. Dong, F. Huang, and C. Xu, "Transformers in computational visual media: A survey," *Computational Visual Media*, 2022.

[210] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *ICCV*, 2021.

[211] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid, "Multiview transformers for video recognition," in *CVPR*, 2022.

[212] J. Yang, X. Dong, L. Liu, C. Zhang, J. Shen, and D. Yu, "Recurring the transformer for video action recognition," in *CVPR*, 2022.

[213] S. Yang, X. Wang, Y. Li, Y. Fang, J. Fang, W. Liu, X. Zhao, and Y. Shan, "Temporally efficient vision transformer for video instance segmentation," in *CVPR*, 2022.

[214] Y. Yang, L. Jiao, X. Liu, F. Liu, S. Yang, Z. Feng, and X. Tang, "Transformers meet visual learning understanding: A comprehensive review," *arXiv*, 2022.

[215] Z. Yang, Y. Wei, and Y. Yang, "Associating objects with transformers for video object segmentation," *NeurIPS*, 2021.

[216] L. Ye, M. Rochan, Z. Liu, X. Zhang, and Y. Wang, "Referring segmentation in images and videos with cross-modal self-attention network," *TPAMI*, 2021.

[217] J. Yin, J. Shen, C. Guan, D. Zhou, and R. Yang, "Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention," in *CVPR*, 2020.

[218] B. Yu, M. Tang, L. Zheng, G. Zhu, J. Wang, H. Feng, X. Feng, and H. Lu, "High-performance discriminative tracking with transformers," in *ICCV*, 2021.

[219] B. Yu, W. Li, X. Li, J. Lu, and J. Zhou, "Frequency-aware spatiotemporal transformers for video inpainting detection," in *ICCV*, 2021.

[220] W. Yu, H. Zheng, M. Li, L. Ji, L. Wu, N. Xiao, and N. Duan, "Learning from inside: Self-driven siamese sampling and reasoning for video question answering," *NeurIPS*, 2021.

[221] Z. Yu and N. Han, "Accelerated masked transformer for dense video captioning," *Neurocomputing*, 2021.

[222] L. Yuan, R. Qian, Y. Cui, B. Gong, F. Schroff, M.-H. Yang, H. Adam, and T. Liu, "Contextualized spatio-temporal contrastive learning with self-supervision," in *CVPR*, 2022.

[223] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving," *Circuits and Systems for Video Technology*, 2021.

[224] ——, "Temporal-channel transformer for 3d lidar-based video object detection for autonomous driving," *T. Circuits and Systems for Video Technology*, 2021.

[225] S. Yun, J. Kim, D. Han, H. Song, J.-W. Ha, and J. Shin, "Time is MattEr: Temporal self-supervision for video transformers," in *ICML*, 2022.

[226] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *ICML*, 2021.

[227] Y. Zeng, J. Fu, and H. Chao, "Learning joint spatial-temporal transformations for video inpainting," in *ECCV*, 2020.

[228] X. Zha, W. Zhu, L. Xun, S. Yang, and J. Liu, "Shifted chunk transformer for spatio-temporal representational learning," *NeurIPS*, 2021.

[229] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, "Scaling vision transformers," 2022.

[230] B. Zhang, J. Yu, C. Fifty, W. Han, A. M. Dai, R. Pang, and F. Sha, "Co-training transformer with videos and images improves action recognition," *arXiv*, 2021.

[231] C. Zhang, M. Zhang, S. Zhang, D. Jin, Q. Zhou, Z. Cai, H. Zhao, X. Liu, and Z. Liu, "Delving deep into the generalization of vision transformers under distribution shifts," in *CVPR*, 2022.

[232] M. Zhang, Y. Yang, X. Chen, Y. Ji, X. Xu, J. Li, and H. T. Shen, "Multi-stage aggregated transformer network for temporal language localization in videos," in *CVPR*, 2021.

[233] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[234] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe, "Vidtr: Video transformer without convolutions," in *ICCV*, 2021.

[235] Z. Zhang and D. Tao, "Slow feature analysis for human action recognition," *IEEE TPAMI*, 2012.

[236] S. Zheng, S. Chen, and Q. Jin, "Vrdformer: End-to-end video visual relation detection with transformers," in *CVPR*, 2022.

[237] H. Zhou, A. Kadav, F. Lai, A. Niculescu-Mizil, M. R. Min, M. Kapadia, and H. P. Graf, "Hopper: Multi-hop transformer for spatiotemporal reasoning," in *ICLR*, 2021.

[238] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong, "End-to-end dense video captioning with masked transformer," in *CVPR*, 2018.

[239] L. Zhu and Y. Yang, "Actbert: Learning global-local video-text representations," in *CVPR*, 2020.

Javier Selva received his BSc in Computer Science from Universitat Politècnica de València in 2015 and got his MSc in Artificial Intelligence from Universitat Politècnica de Catalunya in 2018. Currently, he is a PhD fellow at Universitat de Barcelona. His research interests include learning video and multi-modal representations with self-supervised approaches and better understanding how and what NNs learn.

Anders Skaarup Johansen received his BSc and MSc in Medialogy from Aalborg University in 2017 and 2019 respectively, and currently holds a position as PhD fellow at Aalborg as a part of the Milestone Research Programme at AAU (MRPA). His research interests include object detection, segmentation, and machine learning on edge in dynamic real-world environments.

Sergio Escalera is Full Professor at the Universitat de Barcelona, Distinguished Professor at Aalborg University, and member of the Computer Vision Center. He is vice president of ChaLearn Challenges in Machine Learning. He is Fellow of the European Laboratory for Learning and Intelligent Systems. His research interests include inclusive, transparent, and fair analysis of humans from visual and multi-modal data.

Kamal Nasrollahi is working on a dual position, Professor of Computer Vision and Machine Learning at Aalborg University and Head of Machine Learning at Milestone Systems. He is interested in fair, ethical, and responsible use of technology, specifically machine learning applied to computer vision for topics like object detection, tracking, anomaly detection, and super-resolution.

Thomas B. Moeslund leads the Visual Analysis and Perception lab at Aalborg University, the Media Technology section at Aalborg University and the AI for the People Center at Aalborg University. His research covers all aspects of software systems for automatic analysis of visual data - especially including people.

Albert Clapés received his PhD in Engineering and Applied Sciences from University of Barcelona in 2019. He is currently a postdoc researcher at Aalborg University and a member of Computer Vision Center (UAB). He is interested in machine learning and computer vision, and in particular in multi-modal and explanatory methods for the analysis of human behavior.

Paper D

Seasons in Drift: A Long-Term Thermal Imaging Dataset for Studying Concept Drift

Ivan Nikolov, Mark P. Philipsen, Jinsong Liu, Jacob V. Dueholm, Anders S. Johansen, Kamal Nasrollahi, Thomas B. Moeslund

The paper has been published in the
Thirty-fifth Conference on Neural Information Processing Systems, pp. 1–20, 2021.

The layout has been revised.

1 Abstract

The time dimension of datasets and long-term performance of machine learning models have received little attention. With extended deployments in the wild, models are bound to encounter novel scenarios and concept drift that cannot be accounted for during development and training. In order for long-term patterns and cycles to appear in datasets, the datasets must cover long periods of time. Since this is rarely the case, it is difficult to explore how computer vision algorithms cope with changes in data distribution occurring across long-term cycles such as seasons. Video surveillance is an application area clearly affected by concept drift. For this reason we publish the Long-term Thermal Drift (LTD) dataset. LTD consists of thermal surveillance imaging from a single location across 8 months. Along with thermal images we provide relevant metadata such as weather, the day/night cycle and scene activity. In this paper we use the metadata for in-depth analysis of the causal and correlational relationships between environmental variables and the performance of selected computer vision algorithms used for anomaly and object detection. Long-term performance is shown to be most correlated with temperature, humidity, the day/night cycle and scene activity level. This suggests that the coverage of these variables should be prioritised when building datasets for similar applications. As a baseline, we propose to mitigate the impact of concept drift by first detecting points in time where drift occurs. At this point we collect additional data that is used to retraining the models. This improves later performance by an average of 25% across all tested algorithms.

2 Introduction

Once computer vision algorithms step outside the lab and are deployed in real-life outdoor applications, their performance tends to drop significantly due to conditions changing over time, i.e. concept drift [24, 87, 92]. Concept drift can materialize as gradual, recurring or sudden changes in the visual representation of the scene. Existing datasets, in general, favour coverage of multiple locations [33, 77] for short periods of time [46, 47, 85]. Such datasets are ill suited for exploring long-term effects such as concept drift and algorithms developed on their basis are unlikely to show robustness to long-term phenomena. Research studying concept drift [28, 57], uses synthetic datasets or datasets augmented in order to introduce drift. This does not necessarily completely represent real-world concept drift.

Our work presents a novel real-world dataset covering the 8 months from January to August. This time span means that the dataset encompasses a wide range of weather conditions, human activity, seasonal transitions, and recurring cycles such as weekdays, weekends, mornings and evenings. Along

with the thermal images, timestamped metadata has been gathered. The metadata includes weather data such as temperature, humidity, precipitation, etc. as well as metrics for scene activity level. We use the dataset to study concept drift by exploring contributing factors and demonstrating their effects on algorithmic performance. By publishing the dataset, we seeks to aid the community in evaluating exiting algorithms against a long-term benchmark and in the development of algorithms that show greater robustness to long-term phenomena.

To explore the dataset, two common tasks are chosen, namely anomaly and people detection. These tasks tend to suffer strong performance degradation when exposed to long-term concept drift [79]. Object detection in general or detecting people in particular is a fundamental task involved in many use cases such as autonomous driving [8, 10, 88], tracking [6, 19, 69, 75] and re-identification [26, 41, 42]. Common for many of the use cases is the application of object detection in unconstrained environments and across long spans of time. Anomaly detection, where the goal is to detect unusual behavioral patterns, is another task that is exposed to concept drift. These algorithms must be able to distinguish irrelevant changes due to e.g. concept drift from emergencies such as burglaries or assaults [77], car accidents [40], loitering and suspicious behaviour [91], indoor [27] and outdoor [15, 37, 44] falls.

We select representative algorithms for each task and evaluate their performance across time and in relation to environmental factors. As expected, all models exhibit performance degradation, as the test data diverges from the training set. Temperature and humidity proves to influence the models the most, followed by the change between day and night and the activity level of the scene. On the other hand, variation in precipitation and wind do not influence the performance of the models. In general, methods that learn from solving tasks that consider the entirety of the image are likely to be less impacted by drift, compared to methods that consider small regions or individual pixels [78]. An example could be object detectors vs. autoencoders, where something like brightness is likely to impact the autoencoder's reconstruction significantly, but won't effect the class or position of objects. By including both autoencoders and object detectors we ensure that both ends of this spectrum are covered in our analysis.

Finally, a baseline algorithm is presented to reduce the consequences of concept drift. This algorithm provides additional training data from points in time where concept drift is detected. This baseline is intended to encourage researchers to develop other methods of reducing the impact of concept drift. We believe that our findings on this novel dataset generalize to other environments and use cases, as well as other modalities and therefore will be an example to follow for future definition and collection of datasets. This in turn will help the community getting closer to deploying long-term computer vision algorithms for real-life outdoor applications. The main contributions of

this paper can be summarized as follows:

- The Long-term Thermal Drift (LTD) dataset - the longest-spanning systematically collected thermal dataset comprised of 8 months of video data, containing both timestamp and weather condition metadata;
- In-depth analysis of the correlational and causal relationships between the performance of models and environmental factors;
- A baseline algorithm for reducing the effects of concept drift.

3 Related Work

3.1 Concept Drift Detection

As many systems need to be deployed and work stably for long periods of time and with input data which can change both gradually and suddenly, the presence of drift and ways to deal with it is a topic that has been widely studied. In computer vision it is normally studied by either focusing on specific real-world use cases or synthetically augmenting existing datasets. Real-world cases can be taken from egocentric video [54] or industrial inspection [53]. These cases present both examples of the problem and detection methods, but have limited use outside of the specific environments. Augmented versions of popular datasets such as MNIST and CIFAR can also be used. The works by [57] and [63] focus on methods for detecting data shifts using differences between the training and testing data, utilizing dimensionality reduction and statistical tests like Maximum Mean Discrepancy and Kolmogorov-Smirnov test. The benefit of using synthetically augmented data for testing is that different types of shifts can easily be simulated - from gradual drift to adversarial attacks [28]. But these simulated shifts do not always correspond to real-world ones. Some more robust methods also exist [79], aimed at using real-world drift in wider variaty of use cases. The need for more research into concept drift, paired with a long-term real-world dataset is evident, as the effects from it can limit long term deployment of vision systems [2, 74].

3.2 Datasets

We can separate previous work roughly in two types of use cases - datasets that contain a scenes from a stationary location, like the ones captured from CCTV and surveillance cameras and datasets with constantly changing locations, like the ones specifically directed towards autonomous cars, robots and human egocentric footage. The two types of datasets are used for different tasks, like vehicle and pedestrian detection and environmental segmentation for

Table D.1: Existing urban computer vision stationary and changing location datasets. The *Location* can be either changing denoting moving camera like the ones on self-driving cars or stationary like on surveillance cameras. The *Type* of the datasets can be either RGB, thermal or LiDAR, the *Duration* is the size of the dataset in hours, the *Period* is the capturing time span and the *Metadata* is any additional information

	Name	Year	Type	Duration	Period	Metadata
Stationary	UCSD [50]	2010	RGB	3.1	-	-
	Caltech Pedestrian [13]	2011	RGB	10	-	-
	VIRAT [56]	2011	RGB	29	-	-
	Avenue [47]	2013	RGB	0.5	-	-
	ShanghaiTech [46]	2018	RGB	3.6	-	-
	Surveillance Videos [77]	2018	RGB	128	-	-
	Street Scene [64]	2020	RGB	4	2 summers	-
	ADOC [62]	2020	RGB	24	1 day	-
	AU-AIR [5]	2020	RGB	2	-	Time, Positions
	MEVA [12]	2021	RGB/Thermal	144	3 weeks	GPS, Time
	LTD [55](Paper D)	2021	Thermal	298	8 months	GPS, Day/Night, Weather, Time
Changing	KAIST [32]	2015	RGB/Thermal	43.41	-	-
	CVC-14 [20]	2016	RGB/Thermal	11.8	-	-
	Oxford RobotCar [49]	2017	RGB/LiDAR	-	1 year	GPS, IMU, Day/Night, Weather
	Aachen Day-Night [72]	2018	RGB	-	-	GPS, Day/Night, Weather
	Gated2Depth [23]	2019	RGB/LiDAR	-	-	GPS, IMU, Day/Night, Weather
	Dark Zurich [70]	2019	RGB	-	-	GPS, Day/Night
	ACDC [71]	2020	RGB	-	several days	GPS, Weather
	Ford AV [1]	2020	RGB/LiDAR	-	1 year	GPS, IMU Day/Night, Weather, Time
	Bdd100k [89]	2020	RGB	-	-	Weather, Time

changing datasets [1, 33, 89] and pedestrian tracking and anomaly detection for stationary ones [12, 46, 64]. The changing datasets also benefit from more diverse data coming from different sensors, compared to more image based stationary datasets. Our proposed LTD dataset is directed towards advancing the state-of-the-art in stationary location outdoor urban datasets by providing a longer duration, larger variation and rich metadata. A comparison in Table D.1 shows how the dataset stacks against previous work.

Datasets used for autonomous driving with changing locations [1, 23, 72, 89], which contain multiple modalities like LiDARs, RGB, depth cameras, as well as GPS and IMU data. They also contain data with longer duration from multiple days [71] to a whole year [49]. These datasets also focus on presenting adverse weather conditions, which can be used for domain adaptation and making autonomous driving and robotics application more robust [1, 70, 71].Thermal datasets are less prevalent but still widely used [17, 20]. These moving location car datasets normally do not contain explicit information of their duration, as they are captured from many cars and the data is sampled.

On the other hand stationary location datasets do not contain any information about the period over which they were collected. This combined with the relative short duration of many of the widely used datasets ([13, 45, 46, 50]) makes it impossible for them to be used for studying long-term effects on de-

ployed machine learning solutions. The duration of some of these datasets is taken from the research presented in [62]. Some larger datasets are gathered from internet videos [77], which lack the needed continuity for testing gradual concept drift in the data. More recent datasets have been produced with the goal to capture larger variations in the environments [12, 62], but with a limited scope. The lack of metadata is another problem, limiting the study of factors causing concept drift, as only some of the investigated datasets provide insufficient metadata [5, 12, 68]. Most of the investigated datasets focus on RGB data, with only some containing both RGB and thermal data [12, 33]. However, thermal imaging is better at preserving preserving people's anonymity as it does not capture facial and body detail. This removes the need for post-processing like blurring or pixelating faces to protect personal data [38, 48, 90], which is a crucial requirement for complying with the European general data protection regulations (GDPR).The thermal imaging market has seen significant growth [14] and is forecast to expand even more in the following years [35, 67], which makes it necessary for long-term public thermal datasets to be easily accessible

4 The Long-term Thermal Drift (LTD) Dataset

To address the gaps seen in the stationary surveillance state-of-the-art and to leverage the need for more thermal data, a new dataset is proposed. It consists of thermal videos with resolution 288×384 captured through the period of **8 months** using a Hikvision DS-2TD2235D-25/50 thermal camera [30]. The camera is a long wavelength infrared (LWIR) unit, capturing wavelengths between 8 and 14 μm. Raw data is captured through the day and saved in a mp4 format as 8-bit uncalibrated grayscale videos. A pre-processing algorithm is then run through the data. It first cuts the raw files into days starting from $00:00$ and separates them into folders. Each folder is timestamped with the year, month and day timestamp. The videos for each day are then cut into **2-minute** clips selected every 30 minutes through the day, for a total of **298 hours**. These videos are additionally timestamped with hour and minute timestamp. The starting point of the data is May 2020 until September 2020, together with a second part from January 2021, up until May 2021. This gives the data a large weather variation through the winter, spring and summer seasons. The images were taken on the harbor front in Aalborg, Denmark. The approximate longitude and latitude coordinates are given as $(9.9217, 57.0488)$. We provide the dataset - **https://www.kaggle.com/ivannikolov/longterm-thermal-drift-dataset**, together with the code to extract the necessary data and to reproduce the experimental pipeline **https://github.com/IvanNik17/Seasonal-Changes-in-Thermal-Surveillance-Imaging**.

Some examples of seasonal and day and night variation of the captured

data, together with weather and human activity variation can be seen in Figure D.1. These large variations, together with a total size almost twice as large as other datasets in Section 3.2, allows for studying the effects of concept drift on trained models.

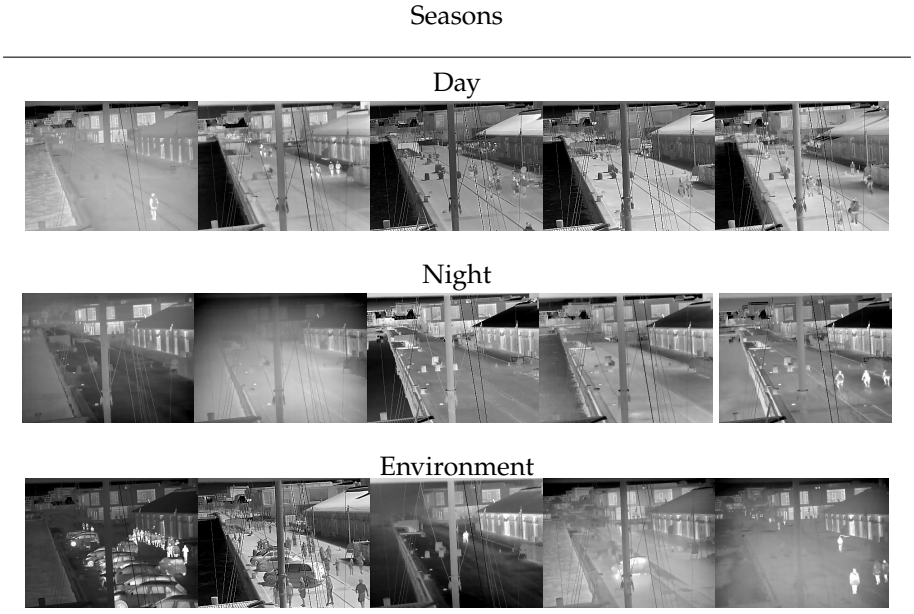


Fig. D.1: Examples of extreme changes in the image data contained in the proposed dataset. From left to right the day and night rows show example changes from data of February, March, April, June and August. The third row shows changes based on weather conditions and human activity.

Figure D.1 depicts issues stemming from the natural thermal data concept drift, such as grayscale inversion in the background and people in different seasons, view limitation and reflections caused by weather like fog, rain, snow, view cluttering from multiple people and vehicles.

4.1 Metadata Analysis

Besides video data we also provide metadata in the form of weather data, gathered using the open source Danish Meteorological Institute (DMI) weather API [34] in 10-minute intervals. The selected properties are - temperature, measured in $[°C]$, relative humidity percentage measured 2m over terrain, accumulated precipitation in $[kg/m^2]$, dew point temperature in $[°C]$ measured 2m over terrain, wind direction in degrees orientation, wind speed in $[m/s]$, both measured 10m over terrain, mean sun radiation in $[W/m^2]$ and minutes of sunshine in the measured interval. These properties are selected, as it is speculated that they would be useful to explain changes in the captured im-

Table D.2: Average metadata for each month. From left - temperature, humidity, precipitation, dew point, wind direction, wind speed, sun radiation and minutes of sunshine in a 10-minute interval.

	Temp. [°C]	Hum. [%]	Precip. $[kg/m^2]$	Dew P. [°C]	Wind Dir. $[degrees]$	Wind Sp. $[m/s]$	Sun Rad. $[W/m^2]$	Sun $[min]$
Jan.	-0.48	90.10	0.01	-1.96	161.91	2.58	23.97	0.90
Feb.	-0.54	85.15	0.01	-2.83	131.00	2.95	51.12	1.42
Mar.	3.75	83.61	0.01	0.93	218.80	3.58	99.35	1.85
Apr.	4.47	97.25	0.13	4.10	126.50	2.97	67.31	2.23
May	10.74	75.46	0.01	6.07	217.32	3.04	256.76	3.66
June	16.36	71.46	0.01	10.57	151.27	2.37	256.46	3.63
July	12.91	75.32	0.01	8.46	268.15	3.97	270.17	3.62
Aug.	16.93	79.17	0.02	12.69	163.18	2.08	197.86	3.15

age data. An overview of the average weather metadata measurements of the dataset can be seen in Table D.2. Temperature and relative humidity have been shown to affect thermal cameras, when detecting surface defects in concrete structures [82], measuring skin temperature changes on athletes [36], getting accurate readings for volcanology [3] and inspecting food [21]. Precipitation and dew point temperature can indicate the presence of rain, fog or high moisture and condensation. These can increase attenuation of infrared light and change the produced camera response [4, 11]. The build-up of moisture can create puddles in the images, which would change the scene reflectivity and reflected temperature [7]. The sun radiation and amount of sunshine can affect the captured images by rapidly changing the intensity of the infrared light. Finally wind speed and direction can cause movement of background parts of the scene like water ripples, ropes, etc., as well as movement of the camera itself.

5 Long-term Performance Experiment

We study the effects of concept drift on six machine learning models - two autoencoders, two object detectors and anomaly detectors. For these experiments only weather parameters not found to have significant correlation to other parameters are considered, namely - temperature, humidity, wind speed, wind direction and precipitation. More information on the correlation between weather parameters is given in the Appendix.

5.1 Data Selection Protocol

In order to keep the experiments and labelling effort manageable, samples across the full data set are selected based on the following protocol. This is done to minimize the number of frames and maximize the variation covered by the selection. For the sampling temperature metadata is used, as it is proven

to directly correlate with changes to thermal images [21, 36, 82]. The protocol can be summarized as follows:

1. Every **2-minute** clip in the dataset is sampled with a frequency of one frame per second, resulting in **120 frames per clip**;
2. Based on the temperature metadata, we select a cold month for the training set and another cold month, a median temperature one, and a warm month for the test set;
3. The training set exists in three variants: coldest day 13th of February, the corresponding week 13-20 of February, and the entirety of February;
4. The test sets consist of data from January (similar cold month), April (month with median temperature), and August (warmest month).

From each of the thus created subsets, a greedy furthest point sampling is used for selecting frames. The frames for each day are sampled by calculating the farthest distances in the 2D feature space of the frame numbering and the temperature. A visual example of the sampling can be seen in the Appendix. The amounts of selected samples vary for the training data depending on the used algorithm. This is further discussed in the next sections.

5.2 Tested Models

Six deep learning models are tested. All six are originally designed to work with RGB data, so their input channel is reduced from 3 to 1, corresponding to a change to the grayscale thermal data. No additional changes were made, as the focus of the paper is not algorithm performance but change in performance over time.

Two of tested models are autoencoders, as representatives for dimensional reduction, noise removal, concept drift detection and anomaly detection methods. Autoencoders are well suited for researching concept drift in long-term datasets, as their reconstruction performance is inherently tightly connected to the training data. The first autoencoder follows a simple fully convolutional architecture with symmetric 5-layer encoder and decoder. The implementation is based on the autoencoder used in a previous work [44]. It is theorized that its simplicity will make it sensitive to concept drift in the input data. The second autoencoder is the latest version of the Vector Quantised Variational Autoencoder (VQVAE2) [65]. This autoencoder uses collections of multi-scale hierarchical discrete tensors, called codebooks, to map its latent space. This gives it more robustness compared to regular autoencoders. The VQVAE2 implementation used here is closely based on [51]. Both autoencoders are trained for 200 epochs.

Two versions of the anomaly detector method MNAD [59] are also tested. They extend traditional autoencoders, by introducing memory-guided normality detection. We look at the typical reconstruction based comparison (MNAD_recon), as well as the prediction approach (MNAD_pred), using the preceding four consecutive frames to predict the future frame. The backbone consists of the U-Net structure, without skip-connections for the MNAD_recon variant. In between the encoder and decoder of U-Net is a memory module, storing prototypical events, concatenated with the original encoder output. The memory is primarily learned during training, but also updates during testing. Both versions are trained for 100 epochs.

Lastly two supervised object detectors are also tested - the YOLOv5 and Faster R-CNN [66]. The chosen hyperparameters for YOLOv5 remain the same as the work in [84], except that the initial learning rate is set to 0.00075 and trained for 200 epochs. The Faster R-CNN is trained for 200 epochs as well with SGD, with initial learning rate set as 0.005, the weight decay as 0.005 and the momentum kept at 0.9. Both object detectors have previously been successfully applied to outdoor thermal imaging [9, 18, 31, 39].

The autoencoders are trained on a NVIDIA GTX1070 Super, the anomaly detectors on a NVIDIA RTX3080 and the object detectors on a NVIDIA RTX2080Ti.

5.3 Drift Algorithmic Performance Analysis

This experiment aims to see how the performance of the selected algorithms changes depending on the variation of the training data.

The training sets for the autoencoders and the anomaly detectors contain 5000 frames per subset, sampled using the method discussed in subsection 5.1, where 20% are used for validation. Performance is reported as the average MSE across every image in each of the three test sets. The performance of the two autoencoders and anomaly detectors is listed in Table D.3. We can see that the MSE for the CAE, VQVAE2 and MNAD_recon increases the farther away the test data goes from the training data. It can also be seen that the larger temporal pool provided for sampling for the weekly and monthly training data helps with keeping the MSE lower through the different months. The MNAD_pred is the only model keeping a consistent performance through the months without any noticeable drift. This is most likely due to the U-Net skip connections being able to reconstruct the background scene with a very low reconstruction error.

For the object detectors, because of the necessary data-labeling a smaller number of images are used for training and testing - both having 100 frames per subset. In addition to these a validation set comprising of 51 images evenly sampled from a previous annotated dataset [44] collected in February 2020 is used. All of the subsets are annotated with bounding boxes around people seen in each frame using the LabelImg open source program [83]. The

Methods	Train Feb.	Test Jan.	Test Apr.	Test Aug.
CAE	Day 5k	0.0096	0.0202	0.0242
	Week 5k	0.0061	0.0167	0.0212
	Month 5k	0.0042	0.0109	0.0147
VQVAE2	Day 5k	0.0051	0.0072	0.0068
	Week 5k	0.0039	0.0066	0.0061
	Month 5k	0.0021	0.0039	0.0035
MNAD Recon.	Day 5k	0.0028	0.0057	0.0069
	Week 5k	0.0065	0.0066	0.0062
	Month 5k	0.0015	0.0041	0.0048
MNAD Pred.	Day 5k	0.0008	0.0007	0.0009
	Week 5k	0.0007	0.0006	0.0007
	Month 5k	0.0007	0.0006	0.0007

Table D.3: Results are reported as the average of the MSE across every frame in the test set. Higher results show worse performance.

Method	Train Feb.	Test Jan.	Test Apr.	Test Aug.
YOLOv5	Day 100	0.8010	0.5390	0.5240
	Week 100	0.7940	0.4540	0.4860
	Month 100	0.7930	0.4860	0.4830
Faster R-CNN	Day 100	0.6760	0.3230	0.3370
	Week 100	0.6740	0.2790	0.3060
	Month 100	0.6400	0.2560	0.3180

Table D.4: Results are reported as the mAP_{50} across every frame in the test set. Lower results show worse performance.

annotations are also part of the LTD dataset. Since the performance of object detector is based on detected bounding boxes, mAP is used to evaluate it. The performance of the object detectors is given in Table D.4. The accuracy of both object detectors, drastically drops in the month of April. To prevent overfitting the smaller amount of training data, we observe the validation and test loss.

As a conclusion from the performance analysis the higher variation provided by sampling from the week and month data, has been translated to better and more stable models in all the tested models. We can still see the effects of the seasonal drift, so additional analysis will be provided in the following sections.

6 Drift Analysis

In this section we look at the possible relations between the observed model performance drift and the changes in the captured metadata. Looking through the data examples given in Figure D.1, two main visual change types are identified - seasonal and day/night. These types can be caused by either changes in the weather conditions, the human activity or a combination between the two. The relation between the model performance metrics and metadata features representing these changes is analysed. As discussed in section 4.1, we choose temperature, humidity, precipitation, wind direction and wind speed as weather data features. For analysing the day/night changes the timestamp data is used to calculate hours of the day, as well as to calculate the sunrise and sunset times [52, 76]. To quantify the activity in the scene the difference between each testing frame and the previous frame from the main dataset is

calculated. The mean value from this difference is selected. To focus only on scene activity everything in the background that moves like the waterfront and the visible ropes and masts is masked out. More information on this can be found in the Appendix.

We choose to use the results only from the models trained on the monthly February data, for easier visualization. The correlation between each of these features and the measured performance metric for each of the methods is first calculated. For the autoencoders and anomaly detectors this is the MSE, while for the object detectors we calculate the F1-score from all images containing people, as it gives a good overview of the precision and recall of the models. Both the basic Pearson's correlation, as well as the more sensitive to non-linear relations Distance correlation [16, 80] are calculated. The statistical significance p-values are also calculated with threshold at 0.05. The calculated correlation r values are given in Table D.5, where those with p-values below the threshold are shown in red.

Table D.5: Correlation between the model's measured performance values MSE and F1-score and the weather, time and scene activity features. Two correlation measures are used - Pearson's (P.C.) and Distance (D.C.) correlation. Measures which do not meet the statistical significance threshold of their p-values are shown in red and marked ✗. The Day/Night features is specified as D./N.

	Measure	Temp.	Hum.	Wind Dir.	Wind Sp.	Precip.	Activ.	D./N.	Hour
CAE - MSE	P. C.	0.679	0.636	0.018 ✗	0.157	0.109 ✗	0.270	0.545	0.166
	D. C.	0.682	0.588	0.158	0.170	0.126 ✗	0.291	0.538	0.287
VQVAE2 - MSE	P. C.	0.381	0.690	0.001 ✗	0.194	0.172	0.217	0.403	0.124
	D. C.	0.347	0.639	0.174	0.201	0.224	0.217	0.382	0.213
MNAD Recon. - MSE	P. C.	0.607	0.672	0.016 ✗	0.173	0.126	0.220	0.509	0.156
	D. C.	0.617	0.629	0.188	0.177	0.155	0.252	0.501	0.273
MNAD Pred. - MSE	P. C.	0.107 ✗	0.277	0.064 ✗	0.152	0.072 ✗	0.677	0.369	0.137
	D. C.	0.231	0.348	0.154	0.172	0.086✗	0.665	0.462	0.312
YOLOv5 - F1-score	P. C.	0.261	0.258	0.102 ✗	0.011 ✗	0.096 ✗	0.124 ✗	0.047 ✗	0.009 ✗
	D. C.	0.293	0.283	0.146 ✗	0.094 ✗	0.135 ✗	0.255	0.113 ✗	0.174 ✗
Faster R-CNN - F1-score	P. C.	0.354	0.456	0.115 ✗	0.135 ✗	0.0124✗	0.199	0.147	0.001 ✗
	D. C.	0.334	0.460	0.228	0.149 ✗	0.065 ✗	0.231	0.163	0.118 ✗

From Table D.5 it can be seen that temperature and humidity have both the largest correlation values to most of the metrics, as well as the most consistently statistically significant results, followed by the scene activity and day/night features. We focus on these four features in the following analysis.

To get a better understanding of not only the correlational, but also causal relations between the models' performance metrics and the chosen features, we look at the Granger causality test [22]. The test only guarantees a predictive causality between variables, but would be able to point out any possible connections. The Granger causality tests the null hypothesis that the past values of one variable do not cause another. The p-value threshold is set to 0.05, below that the null hypothesis can be rejected, with the conclusion that

there is a predictive causality between the variables. As the normal Granger causality test as presented in [73] is used on data with linear relations, we also use the more robust non-linear Neural Granger test [81]. Two best performing versions are used, based on long-short term memory networks (LSTM) and multi-level perceptron (MLP). Both models were trained using proximal gradient descent [58], with $\lambda = 0.002$, ridge regression coefficient 0.01 and learning rate of 0.005. The results from the Granger causality tests are given in Table D.6, where cells shown with green indicate a statistically significant presence of Granger causality and the ones with red - no presence.

Table D.6: Results from calculating linear and non-linear (LSTM and MLP) Granger causality tests. The cells marked with ✓ show positive predictive causality, while cells marked with ✗ show no significant causality.

	Temp.			Hum.			Activ.			D./N.		
	Basic	LSTM	MLP	Basic	LSTM	MLP	Basic	LSTM	MLP	Basic	LSTM	MLP
CAE - MSE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
VQVAE2 - MSE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
MNAD Recon. - MSE	✓	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓	✓
MNAD Pred. - MSE	✓	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
YOLOv5 - F1-score	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗
Faster R-CNN - F1-score	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✓	✓

The results show that the human activity has no predictive causality towards the performance of the models, which combined with the results from the correlation analysis, can point towards a second-hand relation. Our hypothesis is that the change in weather conditions and the day/night cycle are related to the change in human activity. From the other features, temperature has stronger predictive causality towards the autoencoders and anomaly detectors, while humidity and the day/night cycle have a more balanced predictive causality.

Figure D.2 shows the relationship between the features and the model metrics. As a processing step before plotting the temperature and humidity they are first smoothed using a mean filter with a kernel size of 20 and then the MSE is normalized between 0 and 1. This is done as they are not compared, but the trend of their change is visualized. We plot the average values for the training month of February, as a vertical red line, to indicate a "threshold".

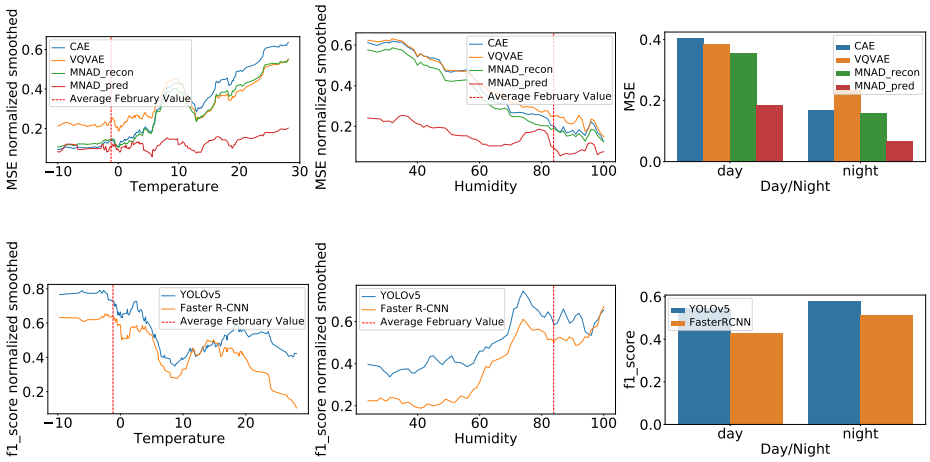


Fig. D.2: Visual representation of the changes of MSE and F1-score for the tested models compared to the temperature, humidity and day/night cycle.

7 Drift Prediction Baseline

As a baseline for exploring and mitigating the effects of concept drift a reference algorithm for predicting drift is presented. We use three strongest features - temperature, humidity and day/night cycle, together with MSE from our convolutional autoencoder (CAE) trained on the February monthly data. The CAE is chosen, as it is the most sensitive to changes in the dataset and is strongly correlated to the performance of all other tested models, except Faster R-CNN. The CAE MSE results from the training data are used together with the chosen features to train two widely used novelty/outlier detection models - isolation forests [43] and one-class SVM [61], available as part of scikit-learn [60]. The isolation forest has 100 base estimators, the one-class SVM has a radial basis function (RBF) kernel and γ of 0.03. We then test the results from each day from the full LTD dataset to detect points where many outliers emerge in both predictors. The first large concentration of outliers in 7 consecutive days is selected, which in our case is 5th of March.

To test if taking in consideration data from the found drift point can help with the performance of the models against concept drift, training data from one week starting after the 5th of March is sampled. The new data is used together with the previous training data from February to retrain the tested models. The results, together with the month results from Table D.3 and D.4 for comparison, are given in Table D.7 and Table D.8. By adding the March data, all tested models achieve better results. We can see that the outlier detection models trained on the CAE MSE, together with the temperature, humidity

and day/night cycle can be used together as a indicator for the amount of drift present in the input data.

Table D.7: The MSE results from the full month in Table D.3, compared to the ones using the new training datasets containing a combination of February and the week in March where drift is detected. Higher results show worse performance.

Methods	Train	Test Jan.	Test Apr.	Test Aug.
VQVAE2	Feb. 5k	0.0021	0.0039	0.0035
	Feb. 5k + Mar. 5k	**0.0020**	**0.0033**	**0.0030**
MNAD Recon.	Feb. 5k	0.0015	0.0041	0.0048
	Feb. 5k + Mar. 5k	**0.0006**	**0.0015**	**0.0025**
MNAD Pred.	Feb. 5k	0.0007	0.0006	0.0007
	Feb. 5k + Mar. 5k	**0.0007**	**0.0005**	**0.0006**

Table D.8: The mAP_{50} Results from the full month in Table D.4, compared to the ones using the new training datasets containing a combination of February and the week in March where drift is detected. Lower results show worse performance.

Method	Train	Test Jan.	Test Apr.	Test Aug.
YOLOv5	Feb. 100	0.7930	0.4860	0.4830
	Feb. 100 + Mar. 100	**0.8690**	**0.6640**	**0.6110**
Faster R-CNN	Feb. 100	0.6400	0.2560	0.3180
	Feb. 100 + Mar. 100	**0.6990**	**0.3910**	**0.3380**

8 Conclusion and Future Work

In this paper we introduced the Long-term Thermal Drift (LTD) dataset spanning 8 months for detecting concept drift in deep learning models. The dataset and the accompanying metadata can be used to document performance degradation as data drifts from the training set. These effects were studied on anomaly and object detection models, as well as autoencoders. It was demonstrated that more diverse training data lowers the effects of concept drift. The performance of the models showed a strong correlational and causal relationship to the change in temperature and humidity. A less pronounced relationship was observed to the day/night cycle and scene activity. Lastly, we showed how the concept drift can be further mitigated by detecting when it starts to manifest and providing additional data to the training process.

The proposed LTD dataset contains a combination of diverse environmental images and granular metadata. The equally spaced long-term data can be used to test the change in performance of deep learning models at different data scenarios - only day or night data, changes between activity in the weekday and weekends, summer and winter scenarios. The influence of weather conditions like rain, snow or fog can also be explored. The possibility of training more robust models and predicting when steps need to be taken, before their performance degrades, is only possible with such long-term sequential datasets.

Possible negative social impacts of such long-term datasets concentrating on a single location is that they can be used to track the habits, interactions and movements of people. We offset this by providing a thermal dataset, which

provides greater protection of people's anonymity than conventional RGB and does not require post-processing for blurring facial features.

The long-term nature of the dataset can also be used, as demonstrated in this paper, to utilize time-series analysis procedures on the outputs from different layers of deep learning models. From simple time-series analysis and forecasting models like Vector Autoregressive (VAR) Models [29] to more complex and data agnostic models like STRIPE [25] or Adversarial Sparse Transformers [86].

We believe that the proposed dataset and the accompanied analysis would help researchers understand the causes for performance drift in models and hence enable easier deployment of long-term solutions in outdoor environments.

References

[1] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1367–1376, 2020.

[2] P. R. Almeida, L. S. Oliveira, A. S. Britto Jr, and R. Sabourin, "Adapting dynamic classifier selection for concept drift," *Expert Systems with Applications*, vol. 104, pp. 67–85, 2018.

[3] M. Ball and H. Pinkerton, "Factors affecting the accuracy of thermal imaging cameras in volcanology," *Journal of Geophysical Research: Solid Earth*, vol. 111, no. B11, 2006.

[4] E. Bernard, N. Rivière, M. Renaudat, M. Péalat, and E. Zenou, "Active and thermal imaging performance under bad weather conditions," 2014.

[5] I. Bozcan and E. Kayacan, "Au-air: A multi-modal unmanned aerial vehicle dataset for low altitude traffic surveillance," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8504–8510.

[6] A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.

[7] D. Bulanon, T. Burks, and V. Alchanatis, "Study on temporal variation in citrus canopy using thermal imaging for citrus fruit detection," *Biosystems Engineering*, vol. 101, no. 2, pp. 161–171, 2008.

[8] L. Chen, N. Ma, P. Wang, J. Li, P. Wang, G. Pang, and X. Shi, "Survey of pedestrian action recognition techniques for autonomous driving," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 458–470, 2020.

[9] Y.-Y. Chen, S.-Y. Jhong, G.-Y. Li, and P.-H. Chen, "Thermal-based pedestrian detection using faster r-cnn and region decomposition branch," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2019, pp. 1–2.

[10] Z. Chen and X. Huang, "Pedestrian detection for autonomous vehicle using multi-spectral cameras," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 211–219, 2019.

[11] J. CORNÉ and U. H. SJÖBLOM, "Investigation of ir transmittance in different weather conditions and simulation of passive ir imaging for flight scenarios," Ph.D. dissertation, MS thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2019.

[12] K. Corona, K. Osterdahl, R. Collins, and A. Hoogs, "Meva: A large-scale multiview, multimodal video dataset for activity detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1060–1068.

[13] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2011.

[14] Y. Dèveloppement, "Thermal imagers and detectors 2020 - covid-19 outbreak impact – preliminary report," http://www.yole.fr/Thermal_Imagers_And_Detectors_Covid19_Outbreak_Impact.aspx, 2020, accessed: 2021-08-11.

[15] F. A. Elshwemy, R. Elbasiony, and M. T. Saidahmed, "A new approach for thermal vision based fall detection using residual autoencoder," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 2, pp. 250–258, 2020.

[16] W. Filho, "Distance correlation," https://gist.github.com/wladston, 2020, accessed: 2021-07-22.

[17] FLIR, "Flir thermal dataset for algorithm training," https://www.flir.com/oem/adas/adas-dataset-form/, 2019, accessed: 2021-09-26.

[18] D. Ghose, S. M. Desai, S. Bhattacharya, D. Chakraborty, M. Fiterau, and T. Rahman, "Pedestrian detection in thermal images using saliency maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[19] Ö. Göçer, K. Göçer, B. Özcan, M. Bakovic, and M. F. Kıraç, "Pedestrian tracking in outdoor spaces of a suburban university campus for the investigation of occupancy patterns," *Sustainable cities and society*, vol. 45, pp. 131–142, 2019.

[20] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López, "Pedestrian detection at day/night time with visible and fir cameras: A comparison," *Sensors*, vol. 16, no. 6, p. 820, 2016.

[21] A. Gowen, B. Tiwari, P. Cullen, K. McDonnell, and C. O'Donnell, "Applications of thermal imaging in food quality and safety assessment," *Trends in food science & technology*, vol. 21, no. 4, pp. 190–200, 2010.

[22] C. W. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica: journal of the Econometric Society*, pp. 424–438, 1969.

[23] T. Gruber, F. Julca-Aguilar, M. Bijelic, and F. Heide, "Gated2depth: Real-time dense lidar from gated images," in *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

[24] M. Guan, C. Wen, M. Shan, C.-L. Ng, and Y. Zou, "Real-time event-triggered object tracking in the presence of model drift and occlusion," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2054–2065, 2018.

[25] V. L. Guen and N. Thome, "Probabilistic time series forecasting with structured shape and temporal diversity," *arXiv preprint arXiv:2010.07349*, 2020.

[26] H. Han, M. Zhou, X. Shang, W. Cao, and A. Abusorrah, "Kiss+ for rapid and accurate pedestrian re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, pp. 394–403, 2020.

[27] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine, "An integrated vision-based approach for efficient human fall detection in a home environment," *IEEE Access*, vol. 7, pp. 114 966–114 974, 2019.

[28] M. A. Hashmani, S. M. Jameel, H. Al-Hussain, M. Rehman, and A. Budiman, "Accuracy performance degradation in image classification models due to concept drift," *Int. J. Adv. Comput. Sci. Appl*, vol. 10, 2019.

[29] J. M. Haslbeck, L. F. Bringmann, and L. J. Waldorp, "A tutorial on estimating time-varying vector autoregressive models," *Multivariate Behavioral Research*, vol. 56, no. 1, pp. 120–149, 2021.

[30] Hikvision, "Ds-2td2235d-25/50," https://us.hikvision.com/en/products/more-products/discontinued-products/thermal-camera/thermal-network-bullet-camera-ds, 2015, accessed: 2021-05-27.

[31] N. U. Huda, B. D. Hansen, R. Gade, and T. B. Moeslund, "The effect of a diverse dataset for transfer learning in thermal person detection," *Sensors*, vol. 20, no. 7, p. 1982, 2020.

[32] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and baselines," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[33] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *CVPR*, 2015.

[34] D. M. Institute, "Dmi api," https://confluence.govcloud.dk/display/FDAPI, 2019, accessed: 2021-05-27.

[35] M. Intelligence, "Ir camera market - growth, trends, covid-19 impact, and forecasts (2021 - 2026)," https://www.mordorintelligence.com/industry-reports/ir-camera-market, 2021, accessed: 2021-08-11.

[36] C. James, A. Richardson, P. Watt, and N. Maxwell, "Reliability and validity of skin temperature measurement by telemetry thermistors and a thermal camera during exercise in the heat," *Journal of thermal biology*, vol. 45, pp. 141–149, 2014.

[37] I. Katsamenis, E. Protopapadakis, A. Voulodimos, D. Dres, and D. Drakoulis, "Man overboard event detection from rgb and thermal imagery: Possibilities and limitations," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–6.

[38] M. Kieu, A. D. Bagdanov, and M. Bertini, "Bottom-up and layerwise domain adaptation for pedestrian detection in thermal images," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 17, no. 1, pp. 1–19, 2021.

[39] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE access*, vol. 8, pp. 125 459–125 476, 2020.

[40] S. K. Kumaran, D. P. Dogra, and P. P. Roy, "Anomaly detection in road traffic using visual surveillance: A survey," *arXiv preprint arXiv:1901.08292*, 2019.

[41] A. R. Lejbølle, K. Nasrollahi, B. Krogh, and T. B. Moeslund, "Person re-identification using spatial and layer-wise attention," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1216–1231, 2019.

[42] H. Li, M. Yang, Z. Lai, W. Zheng, and Z. Yu, "Pedestrian re-identification based on tree branch network with local and global learning," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 694–699.

[43] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.

[44] J. Liu, M. P. Philipsen, and T. B. Moeslund, "Supervised versus self-supervised assistant for surveillance of harbor fronts," in *16th International Conference on Computer Vision Theory and Applications (VISAPP)*, 2021.

[45] Q. Liu, Z. He, X. Li, and Y. Zheng, "Ptb-tir: A thermal infrared pedestrian tracking benchmark," *IEEE Transactions on Multimedia*, vol. 22, no. 3, pp. 666–675, 2019.

[46] W. Liu, D. L. W. Luo, and S. Gao, "Future frame prediction for anomaly detection – a new baseline," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[47] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2720–2727.

[48] C. Ma, N. T. Trung, H. Uchiyama, H. Nagahara, A. Shimada, and R.-i. Taniguchi, "Adapting local features for face detection in thermal image," *Sensors*, vol. 17, no. 12, p. 2741, 2017.

[49] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[50] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1975–1981.

[51] A. McKinney, "Vqvae2 implementation," 2021, last accessed: June 7, 2021. [Online]. Available: https://github.com/vvvm23/vqvae-2

[52] J. Meeus, "Astronomical algorithms," *Richmond*, 1991.

[53] C. Mera, M. Orozco-Alzate, and J. Branch, "Incremental learning of concept drift in multiple instance learning for industrial visual inspection," *Computers in Industry*, vol. 109, pp. 153–164, 2019.

[54] P. Nagar, M. Khemka, and C. Arora, "Concept drift detection for multivariate data streams and temporal segmentation of daylong egocentric videos," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1065–1074.

[55] I. A. Nikolov, M. P. Philipsen, J. Liu, J. V. Dueholm, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Seasons in drift: A long-term thermal imaging dataset for studying concept drift," in *Thirty-fifth Conference on Neural Information Processing Systems*. Neural Information Processing Systems Foundation, 2021.

[56] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*. IEEE, 2011, pp. 3153–3160.

[57] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. V. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *arXiv preprint arXiv:1906.02530*, 2019.

[58] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[59] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 14 372–14 381.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[61] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[62] M. Pranav, L. Zhenggang *et al.*, "A day on campus-an anomaly detection dataset for events in a single camera," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[63] S. Rabanser, S. Günnemann, and Z. C. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *arXiv preprint arXiv:1810.11953*, 2018.

[64] B. Ramachandra and M. Jones, "Street scene: A new dataset and evaluation protocol for video anomaly detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2569–2578.

[65] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," 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. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf

[66] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[67] A. M. Research, "Global thermal imaging camera market by 2030," https://www.globenewswire.com/news-release/2021/08/09/2277188/0/en/Global-Thermal-Imaging-Camera-Market-is-Expected-to-Reach-7-49-Billion-by-2030-Says-AMR.html, 2021, accessed: 2021-08-11.

[68] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, "Multi-timescale trajectory prediction for abnormal human activity detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 2626–2634.

[69] Y. F. Said and M. Barr, "Pedestrian detection for advanced driver assistance systems using deep learning algorithms," *IJCSNS*, vol. 19, no. 10, 2019.

[70] C. Sakaridis, D. Dai, and L. V. Gool, "Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7374–7383.

[71] C. Sakaridis, D. Dai, and L. Van Gool, "Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding," *arXiv preprint arXiv:2104.13395*, 2021.

[72] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8601–8610.

[73] S. Seabold and J. Perktold, "statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[74] T. S. Sethi and M. Kantardzic, "Handling adversarial concept drift in streaming data," *Expert systems with applications*, vol. 97, pp. 18–40, 2018.

[75] G. Shen, L. Zhu, J. Lou, S. Shen, Z. Liu, and L. Tang, "Infrared multi-pedestrian tracking in vertical view via siamese convolution network," *IEEE Access*, vol. 7, pp. 42 718–42 725, 2019.

[76] K. Stopa, A. Kobyshev, Matthias, and H. Bertrand, "Suntime," https://github.com/SatAgro/suntime, 2019, accessed: 2021-07-22.

[77] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[78] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[79] A. Suprem, J. Arulraj, C. Pu, and J. Ferreira, "Odin: automated drift detection and recovery in video analytics," *arXiv preprint arXiv:2009.05440*, 2020.

[80] G. J. Székely, M. L. Rizzo, N. K. Bakirov *et al.*, "Measuring and testing dependence by correlation of distances," *The annals of statistics*, vol. 35, no. 6, pp. 2769–2794, 2007.

[81] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox, "Neural granger causality," *arXiv preprint arXiv:1802.05842*, 2018.

[82] Q. H. Tran, D. Han, C. Kang, A. Haldar, and J. Huh, "Effects of ambient temperature and relative humidity on subsurface defect detection in concrete structures by active thermal imaging," *Sensors*, vol. 17, no. 8, p. 1718, 2017.

[83] Tzutalin, "Labelimg," https://github.com/tzutalin/labelImg, 2015, accessed: 2021-06-06.

[84] Ultralytics, "Yolov5," 2020, last accessed: April 15, 2021. [Online]. Available: https://github.com/ultralytics/yolov5

[85] M. Wang, W. Li, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3274–3281.

[86] S. Wu, X. Xiao, Q. Ding, P. Zhao, W. Ying, and J. Huang, "Adversarial sparse transformer for time series forecasting," 2020.

[87] L. Yang, W. Guo, Q. Hao, A. Ciptadi, A. Ahmadzadeh, X. Xing, and G. Wang, "{CADE}: Detecting and explaining concept drift samples for security applications," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[88] Z. Yang, J. Li, and H. Li, "Real-time pedestrian and vehicle detection for autonomous driving," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 179–184.

[89] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[90] Y. Zhang, Y. Lu, H. Nagahara, and R.-i. Taniguchi, "Anonymous camera for privacy protection," in *2014 22nd International Conference on Pattern Recognition*. IEEE, 2014, pp. 4170–4175.

[91] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, "Anomalynet: An anomaly detection network for video surveillance," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2537–2550, 2019.

[92] I. Žliobaitė, M. Pechenizkiy, and J. Gama, "An overview of concept drift applications," *Big data analysis: new algorithms for a new society*, pp. 91–114, 2016.

Paper E

ChaLearn LAP Seasons in Drift Challenge: Dataset, Design and Results

Anders S. Johansen, Julio C. S. Jacques Junior Sergio Escalera, Kamal Nasrollahi, Thomas B. Moeslund

The paper has been published in the
European Conference on Computer Vision : Workshop on Real-World Surveillance - Tel Aviv, Israel, vol. 13805, pp. 755-769, 2022.

The layout has been revised.

Abstract

In thermal video security monitoring the reliability of deployed systems rely on having varied training data that can effectively generalize and have consistent performance in the deployed context. However, for security monitoring of an outdoor environment the amount of variation introduced to the imaging system would require extensive annotated data to fully cover for training and evaluation. To this end we designed and ran a challenge to stimulate research towards alleviating the impact of concept drift on object detection performance. We used an extension of the Long-Term Thermal Imaging Dataset, composed of thermal data acquired from 14th May 2020 to 30th of April 2021, with a total of 1689 2-minute clips with bounding-box annotations for 4 different categories. The data covers a wide range of different weather conditions and object densities with the goal of measuring the thermal drift over time, from the coldest day/week/month of the dataset. The challenge attracted 184 registered participants, which was considered a success from the perspective of the organizers. While participants managed to achieve higher mAP when compared to a baseline, concept drift remains a strongly impactful factor. This work describes the challenge design, the adopted dataset and obtained results, as well as discuss top-winning solutions and future directions on the topic.

1 Introduction

In the context of thermal video security monitoring the sensor type that is responsible of quantifying the observed infrared-radiation as a thermograph can be split into two groups: sensors that produce relative thermographs and sensors that produce absolute thermographs. Absolute thermographs can correlate the observed radiation directly with temperature, whereas relative thermographs produce observations relative to the "coldest" and "warmest" radiation. In security monitoring contexts the absolute temperature readings produced by an absolute thermograph are not necessary and can potentially suppress thermal details when observing thermally uniform environment. Furthermore the price of absolute thermal cameras are much higher than their relative counterpart.

When performing image recognition tasks the visual appearance of objects and their surroundings is very important, and in an outdoor context that is subjected to changes in temperature, weather, sun-radiation, among others, the visual appearance of objects and their surroundings change quite drastically. This is further expanded by societal factors like the recent pandemic which could introduce mandatory masks. This is known as "Concept Drift" where objects remain the same however the concept definition which is observed through representation changes. While in theory it could be possible to collect

a large enough dataset encompassing the weather conditions, the actors, usually people, within the context also dress and act differently. Furthermore the cost of producing such a dataset would be quite extensive as potentially years worth of data would have to be annotated. Typically deployment of object detectors would have a pretrained baseline, and the model would have to be retrained when the observed context drifts too far away from the training context. The reliability in such a system is questionable as deployed algorithms tend not to have a way to quantify the performance during deployment and extra data would have to be routinely annotated to verify that the system is still performing as expected. To address this issue and foster more research into long-term reliability of deployed learning based object detectors a benchmark for classifying the impact of concept drift could greatly benefit the field.

The ECCV 2022 ChaLearn LAP Seasons in Drift Challenge aims to propose a setting for evaluating the impact of concept drift at a month to month basis and evaluating the impact of concept drift in a weighted manner. The problem of concept drift is exacerbated with limited training data, particularly when the distribution of the visual appearance in the data is similar. To explore the consistency of performance across varied levels of concept drift particularly of object detection algorithms, an extended set of frames were annotated spanning several months. The challenge attracted a total of 184 participants on its different tracks. With a total of 691 submissions at the different challenge stages and tracks, from over 180 participants, the challenge managed to successfully establish a benchmark for thermal concept drift. Top-wining solutions outperformed the baseline by a large margin following distinct strategies, detailed in Sec. 4.

The rest of the paper is organized as follows. In Sec. 2 we present the related work. The Challenge design, which includes a short description of the adopted dataset, evaluation protocol and baseline are detailed in Sec. 3. Challenge results and top-winning solutions are discussed in Sec. 4. Finally, conclusion and suggestions for future research directions are drawn in Sec. 5.

2 Related Work

Popular thermal detection and segmentation datasets, such as KAIST [13] and FLIR-ADAS [24], provide thermal and visible images. The focus of a large part of academic research have been focused on leveraging a multi-modal input [10, 16, 29, 30] or using the aligned visible/thermal pairs as a way to do unsupervised domain adaptation between the visible and thermal [7, 10, 25, 28]. Approaches that leverage the multi-modal input directly typically use siamese style networks to perform modality specific feature extraction, subsequently leveraging a fusion scheme to combine the information in a

learned manner [16, 25, 29], alternatively simple concatenation or addition is performed after initial feature extraction [10, 30]. In contrast, a network can be optimized to be domain agnostic. HeatNet [25] and DANNet [28] leverage an adversarial approach to guide the network to extract domain agnostic features.

It has been proven that in security monitoring contexts fusion of visible and thermal images outperforms any modality alone [14, 17], however in a real-world scenario camera setups tend to be single sensor setups. While thermal cameras are robust to changes in weather and lighting conditions, they still struggle with the change of visual appearance of objects due to the change of scene temperature [6, 8, 9, 15, 17]. Early work [9] leveraged edges to highlight objects, making detection possible robust to the variation when the relative contrast between objects and their surroundings were consistent. Recent studies leverage research in the visible imaging domain, and directly apply it to the thermal domain [6, 17]. Until recently thermal specific detection methods have been a rarity and recently it was proven that contextual information is important to increase robustness to day/night variation [15, 23] for thermal only object detection. By employing a conditioning of the latent representation guided by an auxiliary day/night classification head, the accuracy of day and night accuracy can be significantly increased [15]. Similar increase in performance can also be gained with a combination of a shallow feature-extractor and residual FPN-style connections [8]. Most notably the residual connections are leverage during training to enforce learning of discriminative features throughout the network, and serve no purpose during inference, and as such can be removed.

3 Challenge Design

The ECCV 2022 Seasons in Drift Challenge[1] aimed to spotlight the problem of concept drift in a security monitoring context and highlight the challenges and limitations of existing methods, as well as to provide a direction of research for the future. The challenge used an extension of the LTD Dataset [21] which consists of thermal footage that spans multiple seasons, detailed in Sec. 3.1. The challenge was split into 3 different tracks associated with thermal object detection. Each track having the same evaluation criteria/data but varying the amount of train data as well as the time span of the data, as detailed next.

- ***Track 1 - Detection at day level:*** Train on a predefined and single day data and evaluate concept drift across time[2]. The day is the 13th of February 2020 as it is the coldest day in the recorded data, due to the

[1]Challenge - `https://chalearnlap.cvc.uab.cat/challenge/51/description/`
[2]Track 1 (on Codalab) - `https://codalab.lisn.upsaclay.fr/competitions/4272`

relative thermal appearance of objects being the least varied in colder environments this is our starting point.

- ***Track 2 - Detection at week level:*** Train on a predefined and single week data and evaluate concept drift across time[3]. The week selected is the week of the 13th – 20th of February 2020 - (i.e. expanding from our starting point)
- ***Track 3 - Detection at month level:*** Train on a predefined and single month data and evaluate concept drift across time[4]. The selected month is the entire month of February.

The training data is chosen by selecting the coldest day, and surrounding data as cold environments introduce the least amount of concept drift. Each track aims at evaluating how robust a given detection method is to concept drift, by training on limited data from a specific time period (day, week, month in February) and evaluating performance across time, by validating and testing performance on months of unseen data (Jan., Mar., Apr., May., Jun., Jul., Aug. and Sep.). The February data is only present in the training set and the remaining months are equally split between validation and test.

Each track is composed of two phases, i.e., development and test phase. At the development phase, public train data was released and participants needed to submit their predictions with respect to a validation set. At the test (final) phase, participants needed to submit their results with respect to the test data, which was released just a few days before the end of the challenge. Participants were ranked, at the end of the challenge, using the test data. It is important to note that this competition involved the submission of results (and not code). Therefore, participants were required to share their codes and trained models after the end of the challenge so that the organizers could reproduce the results submitted at the test phase, in a "code verification stage". At the end of the challenge, top ranked methods that pass the code verification stage were considered as valid submissions.

3.1 The dataset

The dataset used in the challenge is an extension of the Long-Term Thermal Imaging [21] dataset, and spans 188 days in the period of 14th May 2020 to 30th of April 2021, with a total of 1689 2-minute clips sampled at 1fps with associated bounding box annotations for 4 classes (Human, Bicycle, Motorcycle, Vehicle). The collection of this dataset has included data from all hours of the day in a wide array of weather conditions overlooking the harborfront of Aalborg, Denmark. In this dataset depicts the drastic changes of appearance of the

[3]Track 2 (on Codalab) - `https://codalab.lisn.upsaclay.fr/competitions/4273`
[4]Track 3 (on Codalab) - `https://codalab.lisn.upsaclay.fr/competitions/4276`

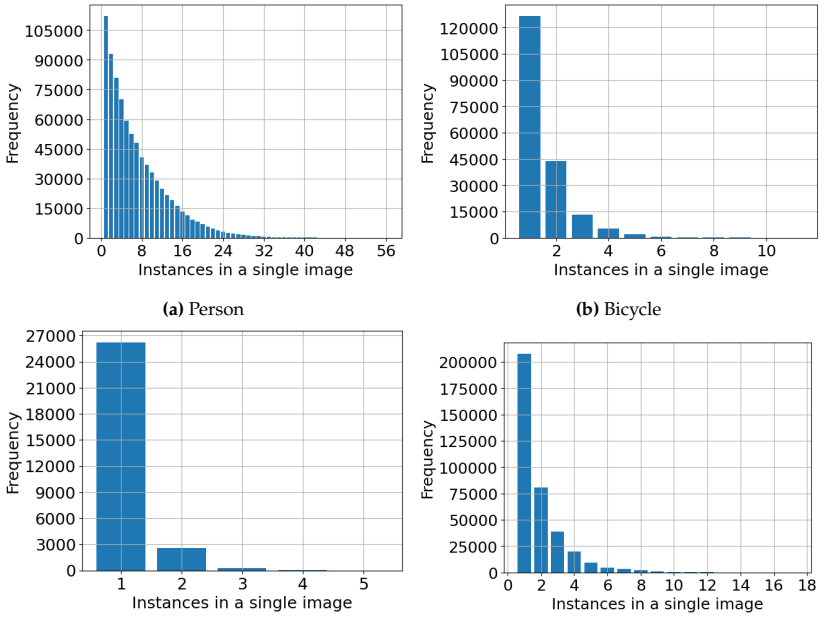
objects of interest as well as the scene over time in a static security monitoring context to develop robust algorithms for real-world deployment. Figure E.1 illustrates the camera setup and two annotated frames of the dataset, obtained at different time intervals.

(a) (b) (c)

Fig. E.1: Illustration of the camera setup (a) and two annotated frames of the dataset, captured at different time intervals (b-c).

For a detailed explination of the datasets weather contents, an overview can be found in the original dataset paper [21]. As for the extended annotations provided with this challenge, we can observe that the distribution of classes is heavily skewed towards the classes that are most commonly observed in the context. As can be seen in Table E.1 the total number of occourances of each class is heavily scewed towards the *Person* class. Furthermore, as can be seen in Figure E.2, each class follows roughly the same trend in terms of the density of which they occur. While the most common for all classes is a single count of the given object present in a given image is 1, the range of occurrences are greater for the *Person* category.

The camera used for recording the dataset was elevated above the observed area, and objects often appear very distant with regards to the camera, in combination with the resolution of the camera most objects appear very small in the image (see Figure E.1). Table E.1 summarizes the amount of objects from each class pertaining to each size category. The size is classified using the same scheme as used in the COCO dataset [19], where objects with areas $area < 32^2$, $32^2 < area < 96^2$ and $area > 96^2$ are considered small, medium and large respectively. The density of object sizes are also illustrated in Figure E.3, where it can be more clearly seen that the vast majority of objects fall within the small category for classes. This holds true for classes *Person*, *Bicycle* and *Motorcycle*, where as the *Vehicle* class more evenly covers all size categories. This is a result of larger vehicles only being allowed to drive in the area closest to the camera.



(a) Person **(b)** Bicycle

(c) Vehicle **(d)** Motorcycle

Fig. E.2: Histogram of object density, across the dataset, density of objects (x-axis) and occurrences (y-axis).

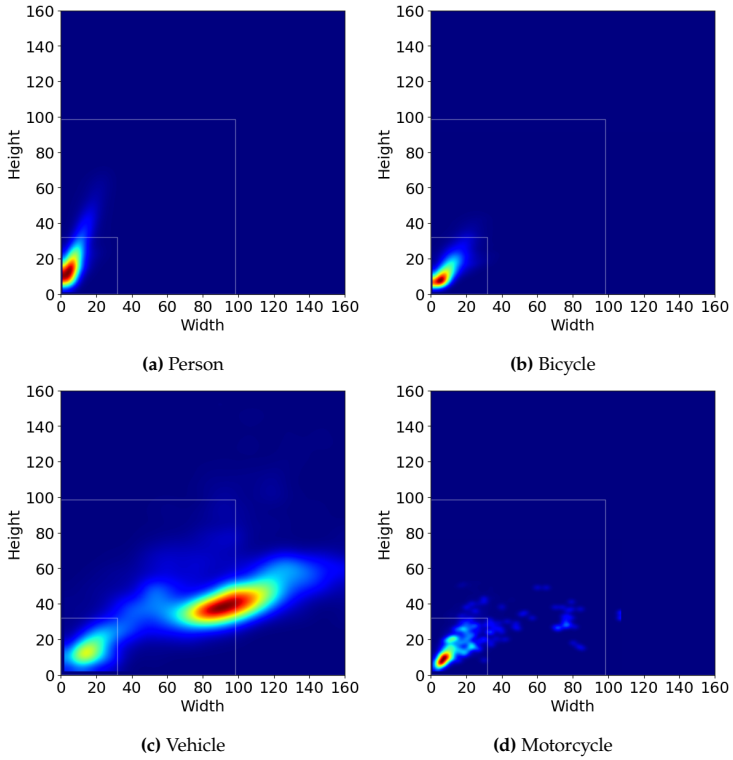
Table E.1: Object frequency observed for each COCO-style size category.

	Class			
Size	Person	Bicycle	Motorcycle	Vehicle
Small	5.663.804	288.081	27.153	113.552
Medium	454	7	0	37.007
Large	176.881	5.192	5.240	550.696
Total	5.841.139	293.280	32.393	701.255

3.2 Evaluation protocol

The challenge followed the COCO evaluation[5] scheme for mAP. The primary metric is, mAP across 10 different IoU thresholds (ranging from 0.5 to 0.95 at 0.05 increments). This is calculated for each month in the validation/test set and the model is then ranked based on a weighted average of each month (more distant months having a larger weight as more concept drift is present), referred to as mAP_w in the analysis of the results (Table E.2). The evaluation is performed leveraging the official COCO evaluation tools[6].

[5] `https://cocodataset.org/#detection-eval`
[6] `https://github.com/cocodataset/cocoapi`



(a) Person **(b)** Bicycle

(c) Vehicle **(d)** Motorcycle

Fig. E.3: Illustration of object size (height×width, in pixels) across the dataset. The white outlines seperate the areas that would be labeled as small, medium and large following COCO standards.

3.3 The baseline

The baseline is a YOLOv5 with the default configuration from the Ultralytics[7] repository, including augmentations. It was trained with a batch size of 64 for 300 epochs, with an input image size of 384×288 and the best performing model is chosen. Naturally, the labels were converted to the normalized yolo format ([cls] [c_x] [c_y] [w] [ht]) for both training and evaluation. For submission on the Codalab platform they were converted back to the ([cls] [tl_x] [tl_y] [br_x] [br_y]) coordinates. The models were all trained on the same machine with 2x Nvidia RTX 3090 GPUs, all training is also conducted as multi GPU training using the pytorch distributed learning module.

[7] `https://github.com/ultralytics/yolov5`

4 Challenge Results and Winning Methods

The challenge ran from 25 April 2022 to 24 June 2022 through Codalab[8] [22], a powerful open source framework for running competitions that involve result or code submission. It attracted a total of 184 registered participants, 82, 52 and 50 on track 1, 2 and 3, respectively. During development phase we received 267 submissions from 17 active teams in track 1, 117 submissions from 6 teams in track 2, and 96 submissions from 4 teams in track 3. At the test (final) phase, we received 84 submissions from 23 active teams in track 1, 55 submissions from 22 teams in track 2, and 72 submissions from 24 teams in track 3. The reduction in the number of submissions from the development to the test phase is explained by the fact that the maximum number of submissions per participant on the final phase was limited to 3, to minimize the change of participants to improve their results by trial and error.

Table E.2: Codalab leaderboards* at the test (final) phase.

Participant	mAP_w	mAP	Jan	Mar	Apr	May	Jun	Jul	Aug	Sep
Track 1 (day level)										
Team GroundTruth*	**.2798**	**.2832**	.3048	**.3021**	**.3073**	**.2674**	**.2748**	**.2306**	**.2829**	**.2955**
Team heboyong*	.2400	.2434	**.3063**	.2952	.2905	.2295	.2318	.1901	.2615	.1419
Team BDD	.2386	.2417	.2611	.2775	.2744	.2383	.2371	.1961	.2365	.2122
Team Charles	.2382	.2404	.2676	.2848	.2794	.2388	.2416	.2035	.2446	.1630
Team Relax	.2279	.2311	.2510	.2642	.2556	.2138	.2336	.1856	.2214	.2235
Baseline*	.0870	.0911	.1552	.1432	.1150	.0669	.0563	.0641	.0835	.0442
Track 2 (week level)										
Team GroundTruth*	**.3236**	**.3305**	.3708	.3502	**.3323**	.2774	**.2924**	**.2506**	**.3162**	.4542
Team heboyong*	.3226	.3301	.3691	.3548	.3279	**.2827**	.2856	.2435	.3112	.4662
Team Hby	.3218	.3296	.3722	.3556	.3256	.2806	.2818	.2432	.3067	**.4714**
Team PZH	.3087	.3156	**.3999**	**.3588**	.3212	.2596	.2744	.2502	.3013	.3592
Team BDD	.3007	.3072	.3557	.3367	.3141	.2562	.2735	.2338	.2936	.3942
Baseline*	.1585	.1669	.2960	.2554	.2014	.1228	.0982	.1043	.1454	.1118
Track 3 (month level)										
Team GroundTruth*	**.3376**	**.3464**	**.4142**	**.3729**	**.3414**	**.3032**	**.2933**	**.2567**	.3112	**.4779**
Team heboyong*	.3241	.3316	.3671	.3538	.3289	.2838	.2864	.2458	**.3132**	.4735
Team BDD	.3121	.3186	.3681	.3445	.3248	.2680	.2843	.2450	.3062	.4076
Team PZH	.3087	.3156	.3999	.3588	.3212	.2596	.2744	.2502	.3013	.3592
Team BingDwenDwen	.2986	.3054	.3565	.3477	.3241	.2702	.2707	.2337	.2808	.3598
Baseline*	.1964	.2033	.3068	.2849	.2044	.1559	.1535	.1441	.1944	.1827

Top solutions are highlighted in bold, and solutions that passed the "code verification stage" are marked with a *.

4.1 The Leaderboard

The leaderboards at the test phase for the different tracks are shown in Table E.2. Note that we only show here the top-5 solutions (per track), in addition to the baseline results. Top solutions that passed the "code verification stage" are

[8]Codalab - `https://codalab.lisn.upsaclay.fr`

highlighted in bold. The full leaderbord of each track can be found in the respective Codalab competition webpage.

As expected, Table E.2 shows that overall better results are obtained with more train data. That is, a model trained at the month level is overall more accurate than the same model trained at the week level, which is overall more accurate than the one trained at the day level. Therefore, the differences in performance improvement when training the model at the month level (compared to week level) are smaller than those obtained when training the model at the week level (compared to day level), particularly when a large shift in time is observed (e.g., from Jun. to Sep.), suggesting that the increase of train data from week to month level may have a small impact when large shifts are observed. This was also observed by the *Team heboyong* (described in Sec. 4.3), which reported to have only used week level data to train their model (i.e., on Tracks 2 and 3), based on the observation that using more data was not improving the final result. This raises an interesting point in that even for winning approaches the variation of the training data is much more important than the amount of training data, a further analysis of what causes the loss of mAP across will be discussed in 4.4.

Table E.3 shows some general information about the top winning approaches. As it can be seen from Table E.3, common strategies employed by top-winning solutions are the use of pre-trained models combined with data augmentation. Next, we briefly introduce the top-winning solutions that passed the code verification stage based on the information provided by the authors. For a detailed information, we refer the reader to the associated fact sheets, available for download in the challenge webpage [9]See footnote 1. Two participants (i.e., *Team GroundTruth* and *Team heboyong*) ranked best on all tracks. Each participant applied the same method on all tracks, but trained at day, week or month level, detailed as follows.

Table E.3: General information about the top winning approaches.

	Top-1 *Team GroundTruth*	**Top-2** *Team heboyong*
Pre-trained model	✓	✓
External data	✗	✗
Data augmentation	✓	✓
Use of the provided validation set as part of the training set at the final phase	✗	✗
Handcrafted features	✗	✗
Spatio-temporal feature extraction	✗	✗
Object tracking	✗	✗
Leverage timestamp information	✗	✗
Use of empty frames present in the dataset	✗	✗
Construct any type of prior to condition for visual variety	✗	✗

4.2 Top-1: *Team GroundTruth*

The *Team GroundTruth* proposed to take benefit of temporal and contextual information to improve object detection performance. Based on Scaled-YOLOv4 [26], they first perform sparse sampling at the input. The best sampling setting is defined based on experiments given different sampling methods (i.e., average sampling, random sampling, and active sampling). Mosaic [1] data augmentation is then used to improve the detector's recognition ability and robustness to small objects. To obtain a more accurate and robust model at inference stage, they adopt Model Soups [27] for model integration, given the results obtained by Scaled-YOLOv4p6 and Scaled-YOLOv4p7 detectors trained using different hyperparameters, also combined with horizontal flip data augmentation to further improve the detection performance. Given a video sequence of region proposals and their corresponding class scores, Seq-NMS [12] associates bounding boxes in adjacent frames using a simple overlap criterion. It then selects boxes to maximize a sequence score. Those boxes are used to suppress overlapping boxes in their respective frames and are subsequently re-scored to boost weaker detections. Thus, Seq-NMS [12] is applied as post-processing to improve the performance further. An overview of the proposed pipeline is illustrated in Figure E.4.

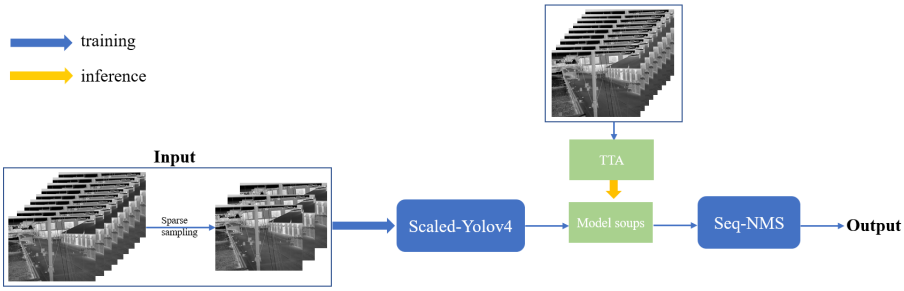


Fig. E.4: Top-1 winning solution pipeline: *Team GroundTruth*.

4.3 Top-2: *Team heboyong*

The *Team heboyong* employed Cascade RCNN [4], a two-stage object detection algorithm, as the main architecture for object detection, with Swin Transformer [20] as backbone. According to the authors, Swin Transformer gives better results when compared with other CNN-based backbones. CBNetv2 [18] is used to enhance the Swin Transformer to further improve accuracy. MMdetection [5] is adopted as the main framework. During training, only 30% of the train data is randomly sampled, to reduce overfitting, combined with different data augmentation methods, such as Large Scale Jitter, Random Crop,

MixUp [31], Albu Augmentation [3] and CopyPaste [11]. At inference stage, they use Soft-NMS [2] and flip augmentation to further enhance the results. An overview of the proposed pipeline is illustrated in Figure E.5. They also reported to have not addressed well the long-tail problem caused by the extreme sparsity of the bicycle and motorcycle categories, which resulted in low mAP for these two categories.

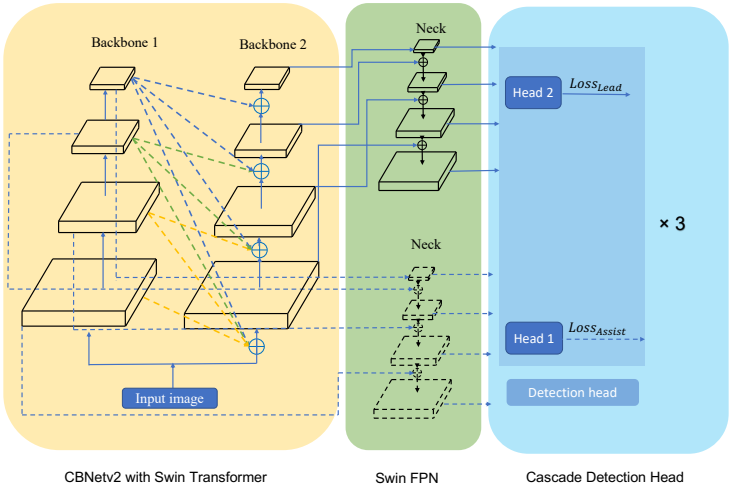


Fig. E.5: Top-2 winning solution pipeline: *Team heboyong*.

4.4 What challenge the models the most?

In this section we analyze the performance of the baseline, Team GroundTruths and Team heboyongs models on the test set. Particularly, we inspect the performance of each model with regards to temperature, humidity object area and object density. Temperature and humidity are chosen as they were discovered that these two factors have the highest correlation with visual concept drift [21]. Additionally, because of the uneven distribution of object densities across dataset, the impact of the object density is also investigated.

Impact of temperature.

can be observed in Figure E.6, as the temperature increases the performance of the model degrades. This is expected as the available training data has been picked from the coldest month and as such warmer scenes are not properly represented in the training data, and as mentioned in 3 this is deliberately done as temperature is one of the most impactfull factors of concept drift in thermal images [21]. The performance of the baseline model shows severe degradation

when compared to the winner and *Team heboyong*, while the performance consistently degrades for all models. Interestingly, *Team heboyong* method is distinctly more sensitive to concept drift with the smaller training set, while the winning solutions seems to perform consistently regardless of the amount of data trained on.

Impact of humidity.

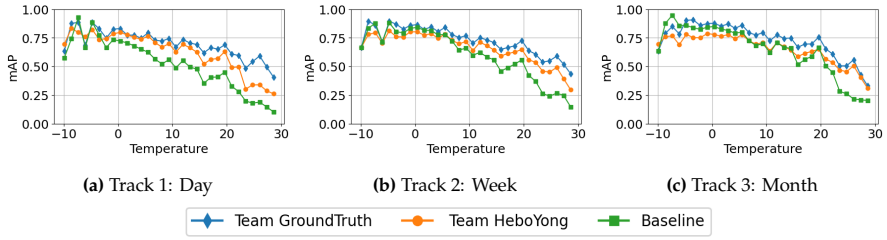
According to the initial paper [21], humidity is one of the most impactfull factors of concept drift, as it tends to correlate positively with the different types of weather. This leads to a quite interesting observation, which can be made across all tracks with regards to the impact of humidity. As can be observed in Figure E.7, the mAP of detectors increases with the humidity across all tracks. This could be because higher humidity tends to correlate with the level of rain-clouds, which would explain partially cloudy being more difficult for the detectors as the visual appearance in the image is less uniform.

Impact of object size.

As would be expected the models converge towards fitting bounding-boxes to the most dominant object size of the training data (see Table E.1). As shown in Figure E.8, the models obtain very good performance on the most common of object sizes and struggle with objects as they increase in size and rarity. In this case the participants see strong improvement over baseline, and also manage to become more robust towards rarer cases. As can also be observed in the figure this problem is increasingly alleviated with the increase of training data.

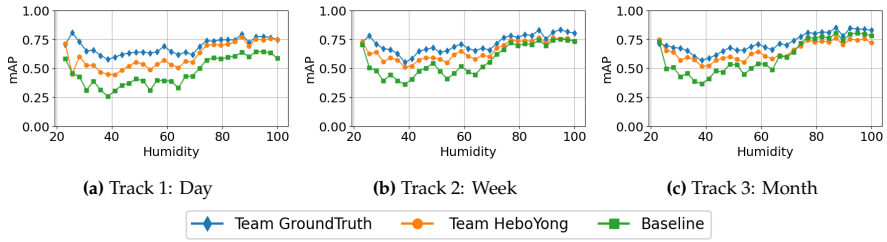
Impact of object density.

As shown in Figure E.2, the density of objects for the majority of the images is towards the lower end, as such one would expect the detectors' mAP to degrade when a scene becomes more crowded and the individual objects become more difficult to detect due to occlusions. However what is observed is the mAP of highlighted methods are consistent as density increases, while the performance across densities also correlate to the amount of training data.



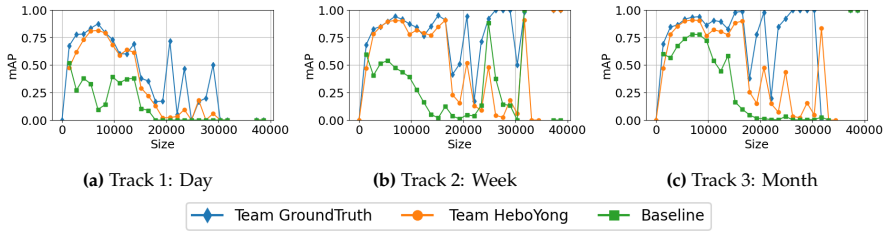
(a) Track 1: Day **(b)** Track 2: Week **(c)** Track 3: Month

Fig. E.6: Overview of performance with samples separated with regards to the temperature recorded for the given frame.



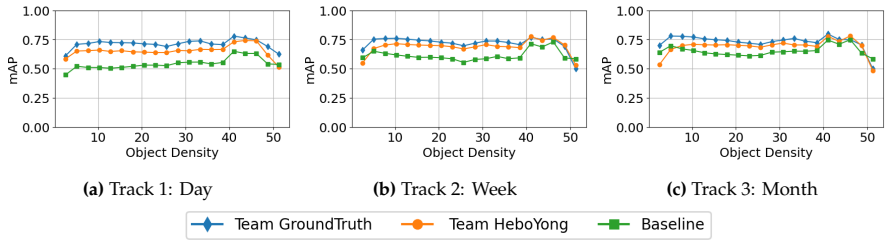
(a) Track 1: Day **(b)** Track 2: Week **(c)** Track 3: Month

Fig. E.7: Overview of performance with samples separated with regards to the humidity recorded for the given frame.



(a) Track 1: Day **(b)** Track 2: Week **(c)** Track 3: Month

Fig. E.8: Overview of performance with samples separated with regards the size of objects bounding-box



(a) Track 1: Day **(b)** Track 2: Week **(c)** Track 3: Month

Fig. E.9: Overview of performance with samples separated with regards to the object density of the frame

5 Conclusions

The Seasons in Drift challenge attracted over 180 participants whom made 480 submissions during validation and 211 submissions for test set and a potential place on the finale leaderboard. While the concept of measuring the impact of thermal drift on detection performance in a security monitoring context is a very understudied field, a lot of people participated. Many of the participants managed to beat the proposed baseline by quite a large margin, especially with limited training data, and achieved more robust solutions when compared to the degradation of the baseline in terms of performance with respect to drift. Allthough great improvements can be observed, the problem of concept drift still negatively affects the performance of participating methods. Interestingly while the winner and *Team heboyong* methods use different architectures, the impact of concept drift seems to transcend the choice of SotA object detectors. This lends merit investigating methods that could condition layers of the network given the input image, and introduce a venue for the model to learn an adaptable approach as opposed to learning a generalized model specific to the thermal conditions of the training context. As can be observed in Figures E.8 and E.9 the size of the observed objects seem to be a more challenging factor than the density of which they occour in. Detection of small objects is a known and well documented problem, and despite the nature of thermal cameras, still persist as an issue in the thermal domain. Further research could be done to learn more scale invariant object detectors or rely entirely on other methods than an RPN or Anchors to produce object proposals.

Acknowledgements

This work has been partially supported by Milestone Research Program at AAU, the Spanish project PID2019-105093GB-I00 and by ICREA under the ICREA Academia programme.

References

[1] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[2] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS - improving object detection with one line of code," in *ICCV*, 2017.

[3] A. V. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020.

[4] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *CVPR*, 2018.

[5] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "MMDetection: Open MMLab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019.

[6] Y.-Y. Chen, S.-Y. Jhong, G.-Y. Li, and P.-H. Chen, "Thermal-based pedestrian detection using faster r-cnn and region decomposition branch," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2019, pp. 1–2.

[7] D. Dai and L. Van Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *ITSC*, 2018.

[8] X. Dai, X. Yuan, and X. Wei, "Tirnet: Object detection in thermal infrared images for autonomous driving," *Applied Intelligence*, vol. 51, no. 3, pp. 1244–1261, 2021.

[9] J. W. Davis and M. A. Keck, "A two-stage template approach to person detection in thermal imagery," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, vol. 1. IEEE, 2005, pp. 364–369.

[10] C. Devaguptapu, N. Akolekar, M. M Sharma, and V. N Balasubramanian, "Borrow from anywhere: Pseudo multi-modal object detection in thermal imagery," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[11] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, and B. Zoph, "Simple Copy-Paste is a strong data augmentation method for instance segmentation," in *CVPR*, 2021.

[12] W. Han, P. Khorrami, T. L. Paine, P. Ramachandran, M. Babaeizadeh, H. Shi, J. Li, S. Yan, and T. S. Huang, "Seq-NMS for video object detection," *CoRR*, vol. abs/1602.08465, 2016.

[13] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *CVPR*, 2015.

[14] X. Jia, C. Zhu, M. Li, W. Tang, and W. Zhou, "Llvip: A visible-infrared paired dataset for low-light vision," in *ICCV*, 2021.

[15] M. Kieu, A. D. Bagdanov, M. Bertini, and A. Del Bimbo, "Task-conditioned domain adaptation for pedestrian detection in thermal imagery," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 546–562.

[16] J. Kim, H. Kim, T. Kim, N. Kim, and Y. Choi, "Mlpd: Multi-label pedestrian detector in multispectral domain," *Robotics and Automation Letters*, vol. 6, no. 4, 2021.

[17] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using yolo," *IEEE access*, vol. 8, pp. 125 459–125 476, 2020.

[18] T. Liang, X. Chu, Y. Liu, Y. Wang, Z. Tang, W. Chu, J. Chen, and H. Ling, "Cbnetv2: A composite backbone network architecture for object detection," *CoRR*, vol. abs/2107.00420, 2021.

[19] 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 *ECCV*, 2014.

[20] 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 *ICCV*, 2021.

[21] I. Nikolov, M. Philipsen, J. Liu, J. Dueholm, A. Johansen, K. Nasrollahi, and T. Moeslund, "Seasons in drift: A long-term thermal imaging dataset for studying concept drift," in *NeurIPS*, 2021.

[22] A. Pavao, I. Guyon, A.-C. Letournel, X. Baró, H. Escalante, S. Escalera, T. Thomas, and Z. Xu, "Codalab competitions: An open source platform to organize scientific challenges," Ph.D. dissertation, Université Paris-Saclay, FRA., 2022.

[23] A. Siris, J. Jiao, G. K. Tam, X. Xie, and R. W. Lau, "Scene context-aware salient object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 4156–4166.

[24] Telodyne, "FLIR AADAS Dataset." [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/

[25] J. Vertens, J. Zürn, and W. Burgard, "Heatnet: Bridging the day-night domain gap in semantic segmentation with thermal images," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8461–8468.

[26] C. Wang, A. Bochkovskiy, and H. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *CVPR*, 2021.

[27] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," *CoRR*, vol. abs/2203.05482, 2022.

[28] X. Wu, Z. Wu, H. Guo, L. Ju, and S. Wang, "Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation," in *CVPR*, 2021.

[29] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon, "Multispectral fusion for object detection with cyclic fuse-and-refine blocks," in *ICIP*, 2020.

[30] H. Zhang, E. Fromont, S. Lefèvre, and B. Avignon, "Guided attentive feature fusion for multispectral pedestrian detection," in *WACV*, 2021.

[31] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *CoRR*, vol. abs/1710.09412, 2017.

Paper F

Who cares about the weather?: Inferring Weather Conditions for Weather-Aware Object Detection in Thermal images

Anders S. Johansen, Sergio Escalera, Kamal Nasrollahi,
Thomas B. Moeslund

The paper is under review in the
MDPI Applied Sciences: New Trends in Image Processing III, 2023

The layout has been revised.

1 Introduction

Deploying thermal image-recognition deep-learning models for long-term analysis of a scene becomes increasingly difficult over time due to concept-drift. Not only does the visual signature of the scene and objects within it change with seasons, but it also changes significantly between day and night. Concept drift is an increasingly researched topic [24, 32], which has focused on identifying distinct concept-drift factors, or assume the presence of distinct distributions. Evaluating and designing methods to combat distinct concept-drift or domain-adaptation is a vital component when deploying computer vision systems in real-world environments. Traditional evaluation methods do not provide accurate description of the impact concept drift has on performance during long-term deployments [32]. Changes in contextual parameters, such as weather conditions, can be somewhat related to the degradation of performance observed with long-term concept drift [32]. For object detection the degradation in performance with relation to temperature and humidity is statistically significant [32]. As these signals are somewhat correlated with change in visual appearance of captured thermal footage, they could be leveraged to guide the model towards learning weather-aware or weather-agnostic representations.

Multi-task learning has become an increasingly popular method for learning generalized image-recognition models [3, 19–21, 24, 33], however mostly focus on using auxiliary branches that are somewhat task-adjacent, where an intuitive connection can be drawn. While each task contributes to shaping the latent-representation to a more generalized representation, which often increases performance for all tasks [20, 33]. Given that the signals induced by the auxiliary tasks are beneficial to achieve a more robust representation, similar approaches could be leveraged to extract and induce a contextually aware signal through auxiliary conditioning.

1.1 Estimating weather

Directly leveraging weather information would require a vision system to directly infer weather conditions from the captured data [2, 7]. By treating it as a classification problem deep learning methods have shown great promise at classifying categories of weather [2, 6, 7]. This shows a weather signal can be somewhat extracted from single-images and categorized into distinct classes. Most weather classification approaches focus on binary classification of distinct weather conditions (i.e. cloudy, sunny, raining, etc.) and lack the granularity observed during long-term deployment. To address this datasets like RFS [16] and MWD [29] propose treating it as a multi-label classification problem to capture the ambiguity between different weather phenomenons and transitive weather conditions [16, 29]. When estimating weather conditions from a single

image, all regions are not created equal [15, 29], thus some methods isolate predetermined regions, such as the sky [2, 45], or leverage region-proposal networks [15, 29], to extract region specific features.

1.2 Adapting to weather

Adverse weather conditions, particularly those which are not present in the training dataset, present a real challenge for performance of deployed computer vision systems which exposed to the weather. Typically approaches to concept drift relies on detecting drift, and then adapting accordingly [12, 31]. With deep-learning based approaches this typically means a general system will be trained to establish a baseline, then subsequently be exposed to unseen data. Depending on the task, an evaluation metric will be used as a method to detect drift [44]. When adapting to weather related drift, some have tried to remove the distracting elements directly [1, 5, 28, 41, 42], train weather-agnostic models by simulating various weather conditions and including that in the training loop if possible [17, 27, 34, 39] or train several models in an ensemble and leveraging a weighted approach to determine the final prediction [13, 25, 37, 40]. Moreover, in situations where an unsatisfactory amount of variation can be captured in the training data; continual-learning [8, 31] or domain-adaptation [12, 31] approaches are often leveraged in an attempt to obtain consistent performance as the visual appearance of the context changes.

1.3 Leveraging metadata for recognition

In recent years, including auxiliary optimization tasks have shown to greatly improve the performance of the down-stream task, whether used as a pre-text task (as often seen with vision transformers [4, 10, 18, 30]), or jointly optimized with the down-stream task [11, 38]. Using auxiliary tasks to guide a primary tasks by introducing aspects that cannot be properly captured in the down-stream task's optimization objective, have shown great promise in improving the performance and generalization of a downstream task [24]. Dependant on the model-architecture and desired purpose of this weather-conditioned representation, it can be leveraged as a constraining parameter which enforces the inclusion of the auxiliary representation directly [24], thereby forcing the network to adjust to be aware of the contextual information induced. Alternatively the auxiliary representation could be seen as purely supplemental information, which potentially consists of redundant elements and as such should only be leveraged to indirectly guide the network.

1.4 Qualitative vs. Quantitative thermal cameras

Thermal cameras work by capturing the amount of infrared radiation object within the scene emit. Though they all aim to capture the same type of information (namely heat), there is two types of thermal cameras. Firstly is qualitative-thermography (sometimes refered to as relative thermal imaging), where the goal is to show the relative differences of infra-red radiation throughout in the cameras field of view. Often used for for inspection and security purposes as they often provide distinct contrast between colder and hotter elements in the field of view regardless of absolute temperature. Secondly is quantitative-thermography (sometimes refered to as absolute thermal imaging), where each sampling point in the field of view is mapped to an absolute temperature measurement. Enabling accurate capture of absolute thermal differences between elements in the scene, and consistent visual response for any thermal signature. In recent years the advances in thermal imaging technology have made the use of thermal cameras increasingly popular, either in isolation or in conjunction with traditional CCTV-cameras. Quantitative thermal cameras could be seen as the ideal solution as they provide essentially the same functionality as qualitative thermal cameras, but with the added benefit of accurate thermal readings. The technology required to construct an absolute-thermograph is significantly more complicated than that of a relative-thermograph and as such are much more costly to produce and purchase. For the purpose of many tasks the absolute temperature readings are redundant for the purpose of the thermal camera, and as such do not justify the cost, thus making qualitative thermal cameras much more commmon in deployed vision systems.

In this paper we will detail a methodology of predicting continuous weather-related meta-variables and provide an overview of the Long-Term Drift (LTD) which contains both object-centric annotations as well as fine-grained weather information for each sample. Further methods decribing how fine-grained weather prediction can be leveraged to condition the network during training to guide the network to become weather-aware. Particularly this will be divided into direct- and indirect-conditioning methods. Lastly this is followed by a discussion of extensive experiments, evaluating the impact of the aforementioned methodology (conditioned on Temperature, Humidity, Time-of-Day), and the impact on performance metrics with respect to the respective weather-conditions. While analysis does not show a direct improvement in accuracy metrics, the analysis shows that auxiliary conditioning in this way does allow the networks to extract and somewhat model the underlying weather signal.

2 Methodology

While more fluid prediction schemes have become available for weather estimation, prediction of weather conditions in literature is still predominantly done in a binary scheme. Using a binary scheme as a conditioning method, implies the assumption that there is a fixed amount of distribution to model. In an uncontrolled environment this is a potentially adverse assumption as unknown variables could induce noise to the signal that would make difficult to distinguish ground-truth close to the bin edges [14, 36]. This is potentially further exacerbated when processing thermal video from cameras with an relative internal thermograph. As detailed in Appendix 1.4, the prevalence of relative thermal cameras makes it a promising modality to investigate, particularly for a real-world context.

To our knowledge the only existing work that performs auxiliary conditioning for task-specific improvements is presented in [24]. They leverage a direct-conditioning approach (detailed in Appendix 2.3) on the KAIST Multispectral Pedestrian Detection (KAIST) dataset and manage to achieve a decrease in Miss-Rate (MR). However the KAIST dataset contains thermal images from an absolute thermal camera, resulting in a fairly similar thermal signature from pedestrians (as seen in Figure F.1).

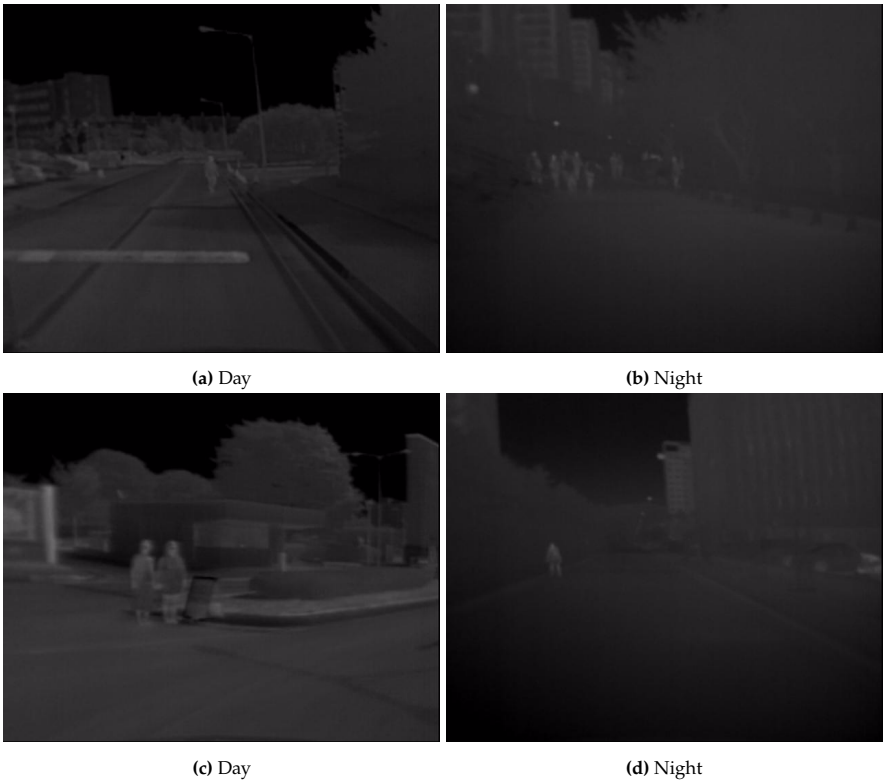
In this section we will detail an overview of the object-centric annotations of the LTD-Dataset and the associated meta-data. Furthermore it will be described how the method proposed in [24] can be adapted for prediction of a continuous auxiliary variable. Finally, we detail the architecture of a direct-conditioning approach (similar to [24]) as well as a indirect-conditioning approach using an State-of-the-Art (SotA) transformer-based model.

2.1 Dataset

In the original LTD dataset benchmark [32] and the subsequent challenge [23] the performance impact concept-drift has on object detector is correlated between the absolute change in mean Average Precission (mAP) across different concept-drift related meta variables (most notably: temperature, humidity, time of day). Subsequently the dataset has been extended with additional object-centric annotations. The dataset was uniformly sampled with a .5 frames per second sample-rate, resulting in over 900.000 images with over 6.000.000 annotated objects.

As can be seen in Figure F.2, the thermal signature of people varies significantly more in the LTD dataset, however that also results in contrast between objects and background varies significantly more.

The objects fall are represented as four classes, namely; person, bicycle, motorcycle and vehicle. The LTD dataset is captured in real-world unconstrained context, and thus i susceptible to associated bias', such as: skewed object



(a) Day **(b)** Night

(c) Day **(d)** Night

Fig. F.1: Examples of thermal images in the KAIST dataset. Where a similar thermal signature of people can be observed at different times of the day, due to the use of quantitative-thermography as well as the limited periods of captured data.

distribution (As seen in Figure F.3a), frames without objects of interest, highly-varied object densities and uneven distribution of weather conditions (As seen in Figures F.4a to F.4c). Furthermore sizes of objects are also affected by the camera being suspended 6 meters above the ground and aimed downwards, resulting in most objects being small (As seen in Figure F.3b and appendix 5). However this is what what could be expected for deployment in a real-world security context.

Furthermore as shown in Figure F.3b while each class has its own unique distribution, the distributions predominantly contain very small objects, with an exception of the vehicle class. This means adds an additional degree of difficulty as most object detectors tend to struggle with smaller objects [4, 26, 35]. Additionally a heatmap with absolute counts of object sizes can be found on Appendix 5.

(a) Day **(b)** Night

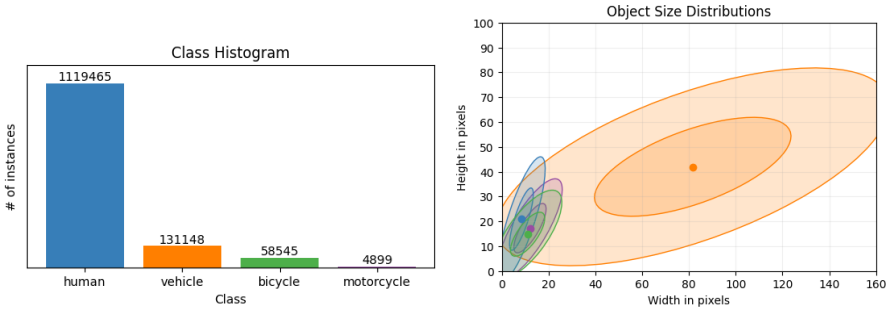
(c) Day **(d)** Night

Fig. F.2: Examples of images with people from the LTD Dataset. Where drastically different thermal signatures for objects can be observed, due to the use of qualitative-thermography, as well as the dataset spanning 9 different months.

2.2 From discrete to continuous meta-prediction

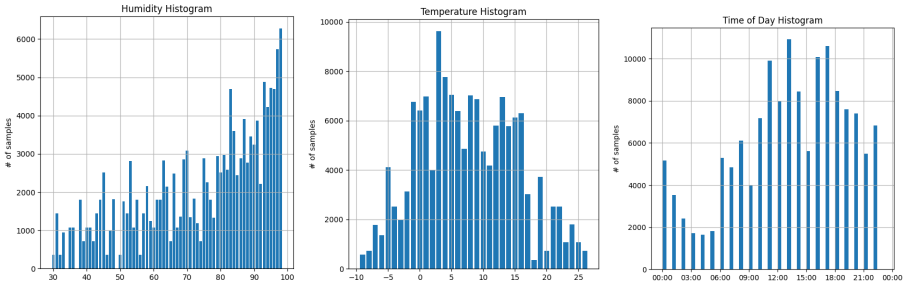
In the KAIST dataset [22], the data falls into two distinct categories, daytime and nighttime, however for real-world deployment the system would observing a gradual change between daytime and nighttime which is not accurately represented by a binary grouping. However in the LTD dataset [32] each clip has an extensive highly granular set of meta-data. This allows us to evaluate the impact of auxiliary task-conditioning in a real-world with more diverse samples.

In [24] they propose guiding the conditioning branch with a binary classification head (Classifying day or night). However, to perform fine-grained continuous weather prediction, the the auxiliary optimization task and loss must be adjusted accordingly. The problem with binary classification is that it treats all false positives equally, regardless of the magnitude of the error. For continuous classifications however, a severity of the miss-classification can be assessed by determining the absolute difference between the prediction



(a) (b)

Fig. F.3: On F.3a the total amount of instances from each of the given classes can be observed, while on F.3b the mean object size can be seen as a dot with additional rings drawn at 1, 2 standard deviations respectively



(a) (b) (c)

Fig. F.4: Histograms showing the distribution of meta-variables across the entire dataset

and the ground-truth. Naively an $L1$-loss can be used to punish/reward the network based on the absolute distance difference. However due to the data being captured by a relative thermal camera, identical visual appearance cannot be guaranteed between calibrations. During capture of the data for the LTD dataset the camera would routinely undergo automatic calibration, resulting in an inconsistent profile over time. This induces a noise signal which could result in the optimization converging towards a global mean rather than an acceptable guess. We combat this by employing an exponential L1 tuned to allow a pre-determined degree of deviation before approaching the values of the primary task loss or losses.

$$L1_e(x, y) \equiv L \equiv \{l_1, ..., l_N\}, l_n \equiv |x_n - y_n|^{\frac{k}{|x_n - y_n|}} \tag{F.1}$$

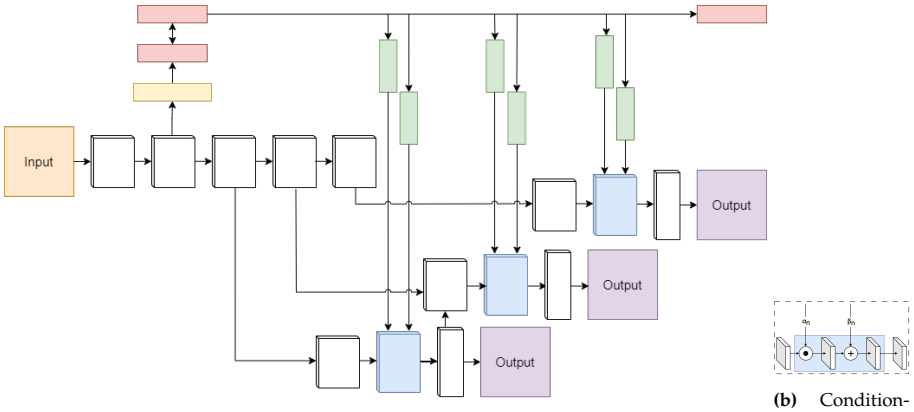
The exponent of the equation is conditioned on the difference between a desired When establishing baseline performance for each model, the minimum,

maximum and standard deviation of the primary loss was noted down for the finale epoch. A corresponding k was then was selected would approach the expected loss values of the primary task the the border of the desired deviation, the resulting weighting of the auxiliary in the optimization process would be approximately equal to that of the primary task, while exponentially increasing when deviating further from the allowed k

Due to the thermal images of the LTD dataset being recorded with a relative thermal camera, visual appearance of a scene might be slightly different, even similar meta-conditions. Thus predicting exact values from visual data would be an ill-posed problem, as any given state inherits some degree of variance from the calibration of the relative thermograph.

2.3 Directly conditioning

In [24] they propose a method of directly conditioning the latent representation of each predictive branches through a conditioning layer. The conditioning element is a part of an auxiliary classification network, which is aimed at predicting whether the given sample belongs to the day-time distribution or the night-time distribution. The latent representation used for this auxiliary prediction is derived from an intermediate representation of the entire image. Thus the representation must be able extract a notion of day and night, which can make the network 'aware' and leverage accordingly.



(a) YOLOv5 with conditioning branch

(b) Conditioning layer

Fig. F.5: YOLO-styled direct-conditioning network. (Figure F.5) and internals of conditioning layer (Appendix 2.3). Red, blue, green, and yellow denote the auxiliary branch-, conditioning layer-, conditioning layer, feed-forward network, and pooling layers respectively.

In the proposed method, the overall mAP, does not improve significantly at higher Intersection over Union (IoU)s, however the weather-conditioned network shows reduction in object MR. By directly conditioning the interme-

diate representation the network is forced directly incorporate the weather information in its semantically rich representation. We employ the original implementation on the YOLOv5 model. As shown in Figure F.5a the standard YOLO architecture is extended with an auxiliary branch, extracted from one of the early stages of the feature extractor, a series of fully connected layers condense the representation which is then fed to a prediction head which predicts a single value which is regressed following the exponential L1 loss described in Appendix 2.2. Individual fully-connected layers feed the representation to the conditioning layer in the different stages of the network, prior to the given stage' prediction head. The conditioning layer (shown in Appendix 2.3), takes in the a set of feature-maps and perform an element-wise multiplication and summation with separated auxiliary representations α_n and β_n respectively.

2.4 Indirectly imposed conditioning

With the recent rise of vision-transformers and their popularity's, they have proven to effectively leverage global reasoning to solve various vision tasks. By calculating an all to all affinity mapping, known as self-attention, between input elements, known as tokens, transformers can effectively relate elements, even when they belong to separate modalities. For classification this is often employed with an additional learnable element, said element is then mapped to an prediction head. The repeated self-attention allows the classification token to extract information from the entire input without directly imposing changes to other inter-token relationships. DETR [4, 26, 46] is a common state-of-the-art transformer-based object detector, and subsequent variants have shown to greatly improve convergence and stability of optimization [26, 46]. By extending the deformable-DETR [46] with a learnable classification token and using the encoding of the classification token to perform prediction of the auxiliary task, namely weather condition prediction. While the optimization could potentially drive the transformer to learn embeddings that are optimized towards affinity with the classification token, the network should be able to disregard weather related embeddings in cases where weather does not provide any significant optimization benefit. Unlike the directly-imposed approach the network could learn to dynamically disregard regions of the image that do not provide contextual information.

Inspired by the use of a [CLS] token in the original BERT [9] paper, we include an additional token with every input sample which is propagated through every encoder layer of the transformer. This way global information from a given sample can be continuously aggregated in a single representation. Prior to reaching the decoder layers, the [CLS] token is seperated and passed to an auxiliary branch (as seen in Figure F.6). The auxiliary branch consists of a series of fully connected layers (sizes; 512, 512, 1) acting as a mapping from weather token to a single value which can be regressed.

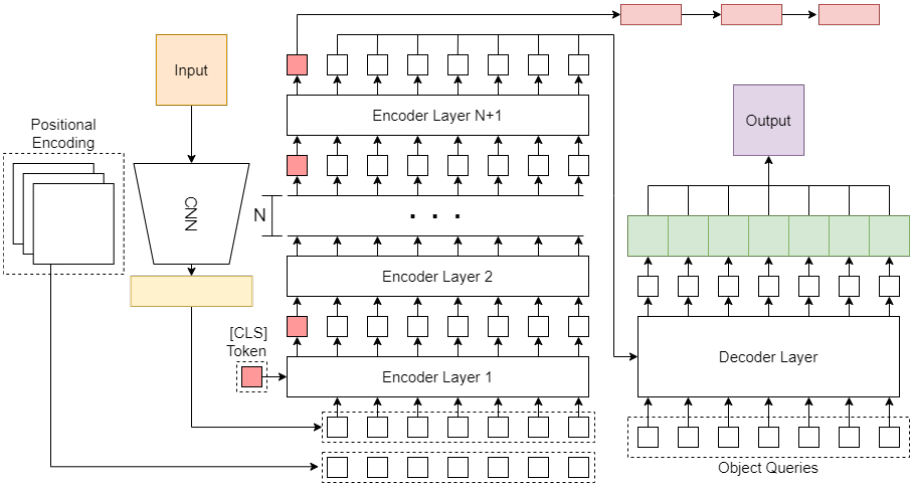


Fig. F.6: DETR-style transformer network with indirect conditioning. Red, light-red, blue, green, and yellow denote the weather-token, auxiliary branch-, feed-forward network, and embedding layers respectively

3 Results

3.1 Experimental setting

To achieve the performance capabilities described in their respective papers and implementations. Since none of the models contained a thermal variant, they were all trained from scratch which required increased training time in order to expect convergence, as "standard" configurations implemented loading an image-net pretrained feature extraction network. As such we set the maximum allowed epochs to 250 for all models. The batch-size for all models were also set comparatively to 8 per GPU, resulting in N total iterations. All models were trained with the same pytorch environment (torch 1.7, torchvision 0.8.1) on two Nvidia RTX 3090 cards. Class-wise losses were weighted to reflect the frequency of each class in their respective subsets. The complete dataset was split into 3 roughly equal parts, where two were employed for training and validation, and the third test set remains hidden to allow for future challenges similar to [23]. Further information data availability is described in the 'Data Availability section at the end of the paper.

3.2 Evaluating weather conditioning

Due to the auxiliary branch being trained in a supervised manner, it has to be exposed to the variaty observed in the training set, as such all clips in the dataset was evenly distributed amongst an equally sized training-, test- and

validation-set. Because this potentially allows a naive approach to generalize easily due to the inclusive of the full variation present in the dataset, the proposed method is compared to an equally trained naive approach, without the auxiliary meta-prediction branch.

To evaluate the potential impact of each of the three meta-conditions (namely; Temperature, humidity and time of day) and the potential improvement, each model was trained naively (i.e. training loop described in the respective paper) as well as with the auxiliary conditioning branch (direct- and indirect- conditioning for YOLO- and DETR-variants respectively). To allow for fair optimization each model is trained for the same amount of epochs as their respective baseline.

Likewise we observe the performance when compared to temperature and object size to investigate if any categories potentially suffer in order to reach a more general improvement of the system. While these correlations might not be intuitively tied to weather, the latent representation learned could inadvertently favor certain aspects of the object distributions.

Because conditions are quantified using different metrics the groundtruth ranges vary significantly. To normalize their representation the values are being remapped so that the observed values fall roughly within the range $[-2, 2]$. This range is chosen to avoid the network having to also learn a mapping between arbitrary ranges, while keeping in line with the normalization done internally in the networks, which is done to avoid unstable variances in the activations [4, 10, 35].

3.3 Accuracy

Table F.1 details the overall mAP and MR for all of their models across validation set. This is used as a metric of overall object detection performance similarly to what is commonly done for other object detection datasets, and to retain a fair comparison with the original LTD dataset evaluation [32]. Additionally Table F.2 details the mean Average Error (mAE) of auxiliary prediction branch, as well as the Standard Deviation (St.Dev.) of the the prediction error. This is listed to provide insight into the performance of the auxiliary branch.

As can be seen in Table F.1 the baseline models which are naively trained without any auxiliary guidance tend to perform better on primary task metrics (mAP), however weather-conditioned variants (particularly temperature variants) display reduced miss-rates, indicating that while their accuracy is generally lower, they recognize more objects than the baseline-counterpart.

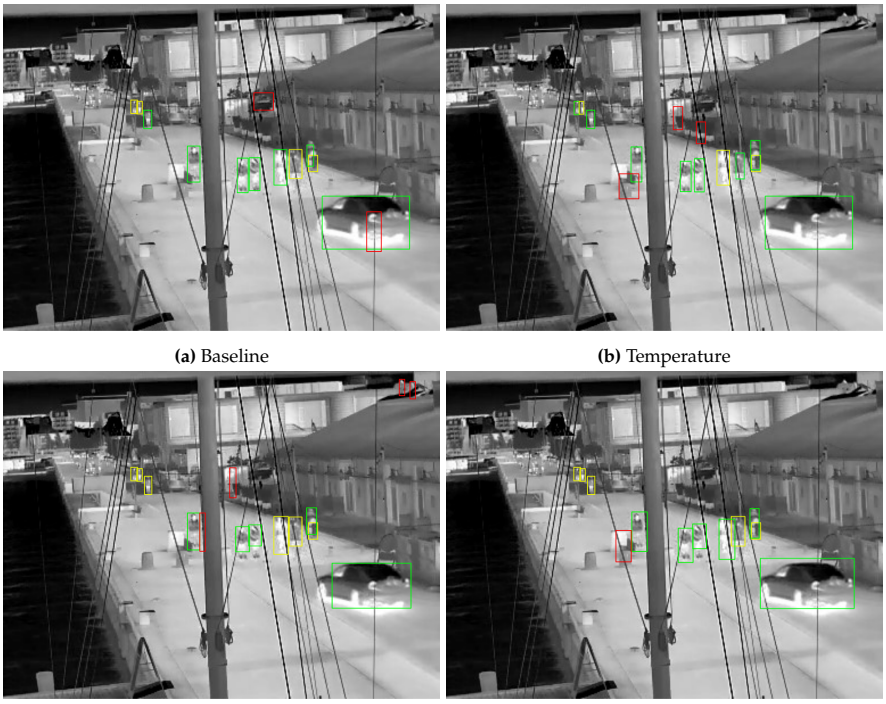
3.4 Accuracy compared to weather

To evaluate the impact of the conditioning branch on the performance with respect to the different weather conditions used for auxiliary prediction and

Model	mAP_{voc}	mAP_{coco}	mAP_L	mAP_M	mAP_S	MR
YOLOv5 (Baseline)	**0.604**	**0.465**	**0.825**	**0.640**	0.491	0.342
YOLOv5 (Pretrain)	0.600	0.454	0.831	0.621	0.489	0.324
YOLOv5 (Temp.)	0.584	0.410	0.796	0.590	0.468	0.322
YOLOv5 (Hum.)	0.493	0.293	0.675	0.560	0.268	0.357
YOLOv5 (ToD)	0.549	0.439	0.805	0.566	0.431	0.356
DN-DETR Baseline	0.378	0.348	0.123	0.344	0.563	0.481
DN-DETR (Temp.)	0.225	0.148	0.100	0.190	**0.682**	0.389
DN-DETR (Hum.)	0.191	0.132	0.100	0.160	0.671	0.415
DN-DETR (ToD)	0.219	0.142	0.00	0.169	0.661	0.410
Def. DETR Baseline	0.332	0.202	0.005	0.051	0.637	0.383
Def. DETR (Temp.)	0.297	0.184	0.001	0.045	0.620	**0.351**
Def. DETR (Hum.)	0.213	0.114	0.000	0.020	0.517	0.416
Def. DETR (ToD)	0.289	0.178	0.001	0.040	0.619	0.395

Table F.1: In this table the mean Average Precission (mAP), and Miss-Rate (MR) of direct- (YOLOv5) and indirect-conditioning (DETR) variants are detailed. Highlighted with **bold** is the best performing across all models and highlighted with underline is the best performing model for a given architecture. mAP_{VOC} denotes mAP where IoU is atleast 0.5, mAP_{COCO} denotes mAP at varying IoUs (i.e. {0.50, 0.55, 0.60, . . . , 0.95}). mAP_L, mAP_M and mAP_S denote mAP of objects with $\{area < 32^2, area > 32^2 < 96^2$ and $area > 96^2\}$ respectively.

optimization, Figures F.9 to F.11 detail the the relation between mAP and the three meta-variables chosen, (namely Temperature, Humidity and time of day). Visual examples to accompany the accuracy overview of Tables F.1 and F.2, can be seen in Figures F.7 and F.8 (Ground truth labels and the image without bounding boxes can be found Figure 12), while visualizations of accuracy with respect to the different weather variables can be seen in Figures F.9 to F.11.



(a) Baseline **(b)** Temperature

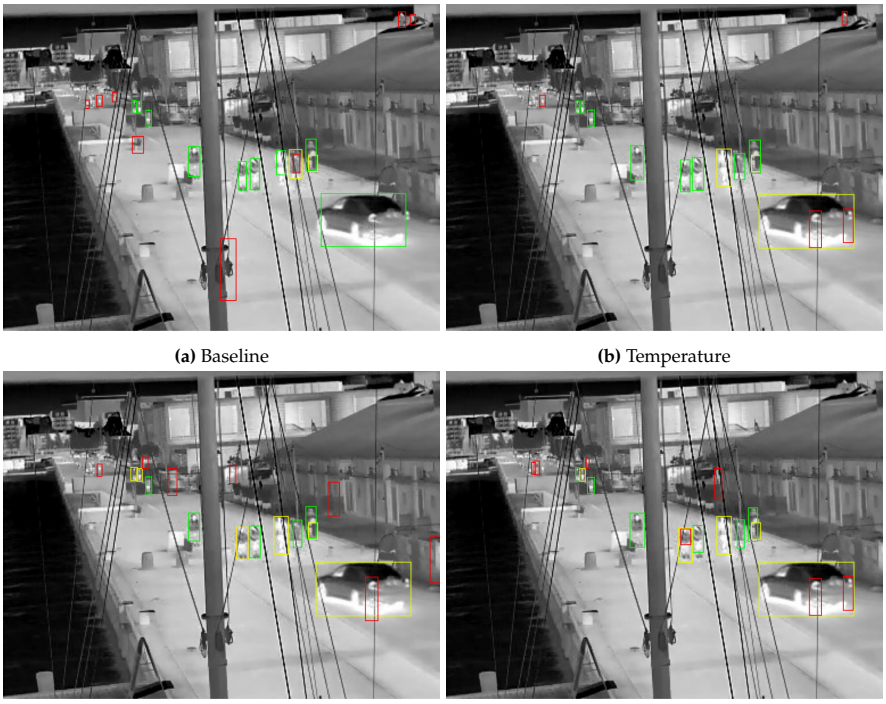
(c) Humidity **(d)** Time of Day

Fig. F.7: Example of Direct-Conditioning performance for each conditioned model. Bounding boxes marked in green, red, yellow are considered True Positives, False Positives, and False Negatives respectively.

In Figure F.9 it can be observed that training models with an temperature-focused auxiliary branch, does not change the performance of said model in any significant way (other than generally lowered mAP). It can be seen that all models follow a curve that is similar to the distribution of samples seen in Figure F.4b, it can be expected that this is happening as the models optimization is simpler when regressing to the mean of the dataset. In addition it can be observed that the indirect-conditioning method is generally more agnostic to variation in the meta variable. Similarly to the temperature focused auxiliary branch, humidity- and time-of-day-conditioning does not seem to improve

	Model	MAE	Std.
Dir.	Temperature	7.1	3.7
	Humidity	18.9	9.4
	Time of Day	7.3	7.1
Indir.	Temperature	5.1	2.9
	Humidity	15.3	8.9
	Time of Day	8.3	7.9

Table F.2: Accuracy of the predicted auxiliary prediction value, $Dir.$ and $Indir.$ denotes the direct- and indirect-conditioning models respectively, while the model row denotes the variant used.



(a) Baseline **(b)** Temperature

(c) Humidity **(d)** Time of Day

Fig. F.8: Example of Indirect-Conditioning performance for each conditioned model. Bounding boxes marked in green, red, yellow are considered True Positives, False Positives, and False Negatives respectively.

overall performance of the models. However interestingly the models seem to be generally agnostic to the distribution of samples (shown in Figures F.4a and F.4c). This indicates that the model has trouble extracting meaningful information with regards to the auxiliary optimization task. This is also present in Table F.2 where it can be seen that the networks have difficulty with accurately predicting their respective weather condition (specifically humidity and time-of-day), whereas temperature prediction is rather accurate, and falls close to the acceptable deviation of the $L1_e$ loss.

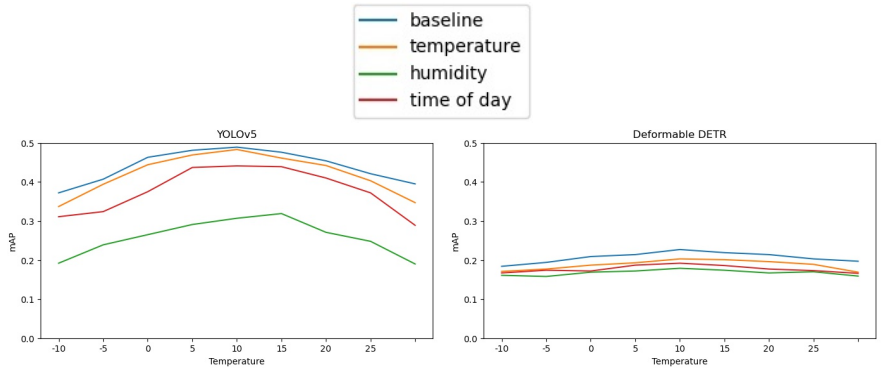


Fig. F.9: Accuracy of models with regards to temperature

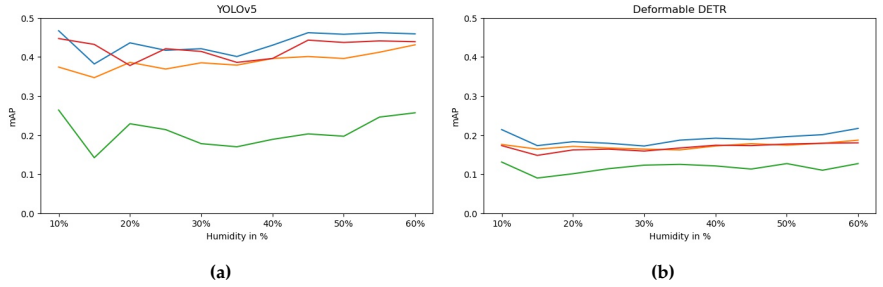


Fig. F.10: Accuracy of models with regards to humidity

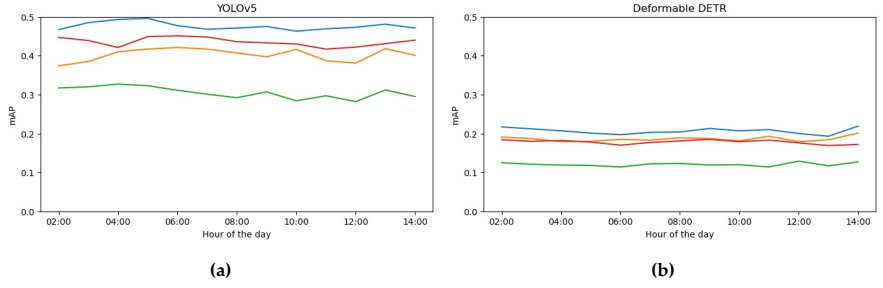


Fig. F.11: Accuracy of models with regards to the time of day

4 Discussion

While previous work have shown that not only is a traditional Convolutional Neural Network (CNN) able to predict weather categories, which in some contexts can help guide the network to be somewhat aware of the distribution a given sample belongs to and adjust accordingly. While this has not shown to increase the accuracy in terms of mAP, it has been shown to decrease false negative predictions. For thermal images with significant concept drift, whether be continual or cyclical, could leave in artifacts which would look appropriate for a given object in one distribution however would be a undesired for another. Intuitively during training the model would either adjust to over-predict (i.e. increased false positives), or under-predict (i.e. increased false negatives) when concept drift is occurs. Essentially the model is tasked with learning an unknown set of distributions, and optimize toward learning to recognize patterns common to the mean of the cumulative distribution. Therefore one could hypothesise that guiding the network towards being aware of a variable correlated with the observed concept drift would allow the network to potentially establish connections between the conditioning representation and the semantic representation used for object prediction.

While it can be observed in Table F.1, the mAP scores do not improve over baseline when conditioned with the auxiliary branch, however the change in MR indicates that the auxiliary branch is enforcing a signal it relates to the auxiliary task. Particularly the temperature conditioned variant manages to detect objects which the baseline fails to detect, however the weather conditioned method also produces an increase in false-positives. Because the visual appearance changes are gradual resulting in a lower accuracy, potentially the weather conditioned learns a more varied representation of given objects which allows it to detect more objects at the cost of false activation's in other places. Additionally one thing that can be observed (in Figure F.9 is that the transformer-based model performs significantly more uniformly across across temperatures, however it is not certain that this is entirely an aspect of the auxiliary predictive branch, or the nature of transformers input dependant attention. Another surprising detail can be found in Table F.1, which shows the DETR-variants in general seem to work really well on small objects which is counter to what is observed in the original and subsequent papers [4, 26, 46]. Intuitively it could be reasoned that the reason for that partially lies in the decoder module, which has a fixed amount of learn-able query tokens, which naturally would converge towards spatial and latent features that are the most prominent, i.e. the person class. Initially an experiment was conducted with regards to the amount of queries to produce, and while increasing them drastically(300->600) would improve performance by 0.3%, however the performance would increase significantly as well. The increased

amount of query tokens could have been kept as a baseline. However, for the sake of keeping baseline models (i.e. YOLOv5 and Deformable DETR) somewhat comparable with other work, the hyperparameters described in their respective repository and paper was kept.

In previous work, namely [24], evaluated performance on a dataset which had two distinct thermal distributions (day and night clips). Our approach assumed that a more continual representation could be learned, however perhaps this cannot be learned fully without an additional proxy that enforces a strict set of conditions forcing formation of distinct distributions.

The appearance shift induced when the thermal camera calibrates to adjust the internal thermograph, perhaps induces a noise into the signal making it difficult to learn a robust approximation of the signal. It could be the visual noise induced partially obfuscates clear delineations between visual groups of visually similar samples, resulting in a regression to the mean being the simplest convergence or perhaps the optimal solution for the downstream task. In such a situation naively training without the auxiliary branch would be the optimal solution if the goal is simply to optimize accuracy. While the auxiliary task does seem to induce noise, both methods (direct- and indirect-conditioning) seem to seem to also somewhat guide the network towards containing a more continual representation, as seen by the reduction in MR. Perhaps trying to constructing distributions as a series of k overlapping distributions, and leveraging a model-soup style approach [43] could provide a more distinct learning of each sub-distribution while still achieving a generalized model of all distributions.

An alternative solution to the regression approach could be a smooth classification approach, where the prediction is considered a smooth positive if it predicts a value within a pre-determined bin-size for each ground-truth number.

5 Conclusion

Thermal concept drift poses a challenging hurdle to overcome when deploying object recognition tasks. Drawing from contextual clues that impact the visual appearance of the scene. Using auxiliary metrics to condition a network directly or indirectly, does not seem to improve the overall performance of the system with regards to mAP, however it does result in a consistent decrease in MR. While not resulting in a direct improvement this shows that a signal can be extracted from the conditioning meta-variable, which can guide the representations learned. The difficulty in accurately modelling the objects across thermal signatures seems to similarly to a naively trained baseline prefer representations that favor the most frequent representations, and as such could be simply seen as a regression to the mean, however due to the networks

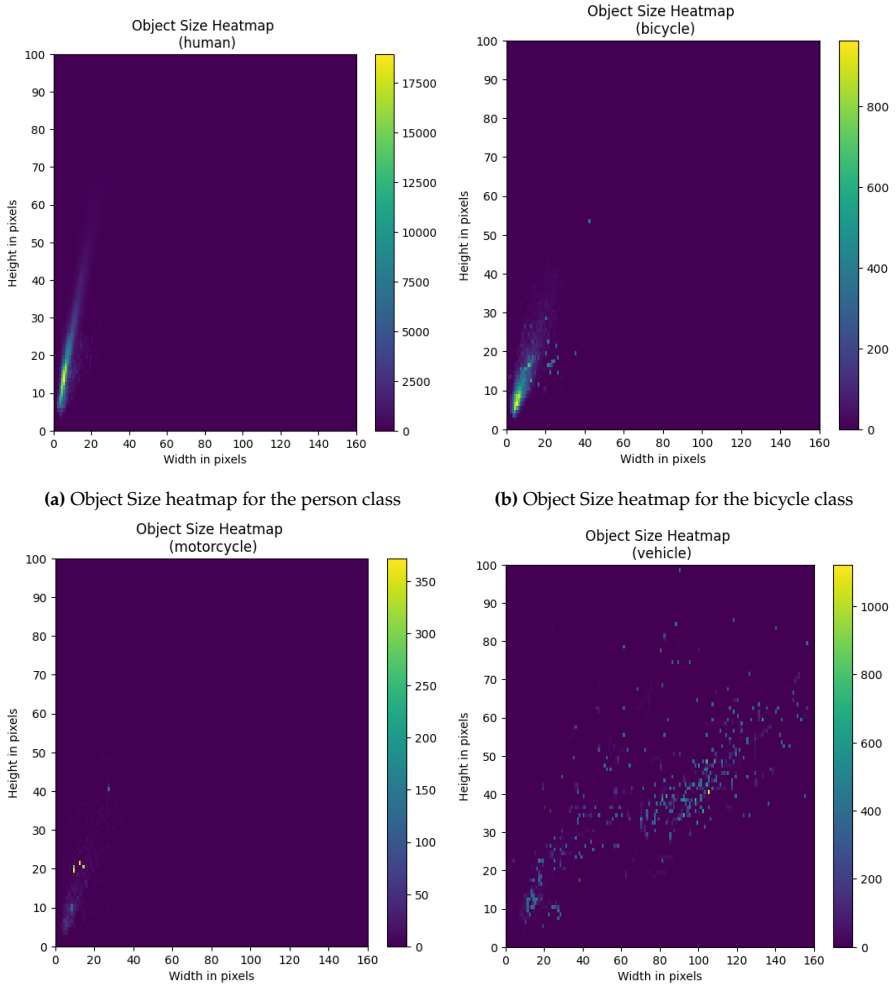
consistently being able to extract a signal related to the auxiliary task, it could imply that deliberately splitting the data into a set of K distributions based on a combination of meta variables or visual appearance could provide more stable guidance.

Appendix

Additional Dataset Figures

(a) (b)

Fig. 12: The example image used in Figures F.7 and F.8 without without bounding boxes(Figure 12a) and with bounding boxes(Figure 12b). In Figure F.7 green, yellow and red refer to person, bicycle and vehicle classes respectively



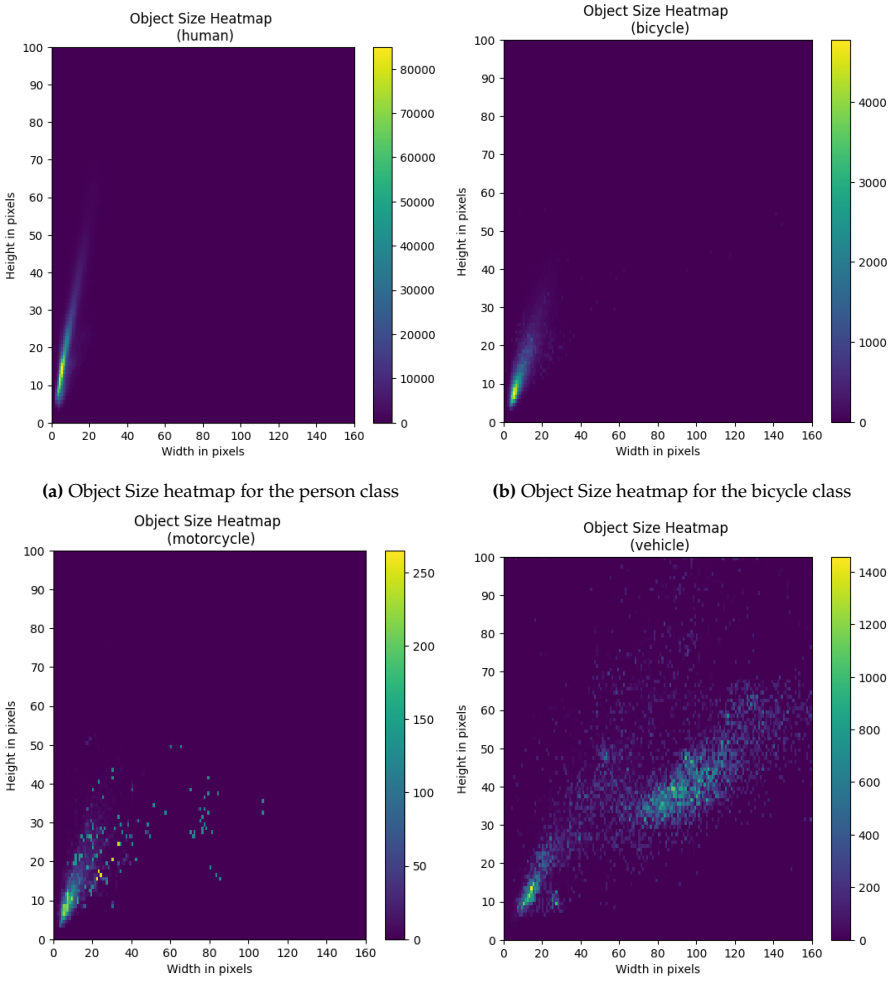
(a) Object Size heatmap for the person class

(b) Object Size heatmap for the bicycle class

(c) Object Size heatmap for the motorcycle class

(d) Object Size heatmap for the vehicle class

Fig. 13: Figures 13a to 13d show a detailed heatmap of the object size distributions for each class individually



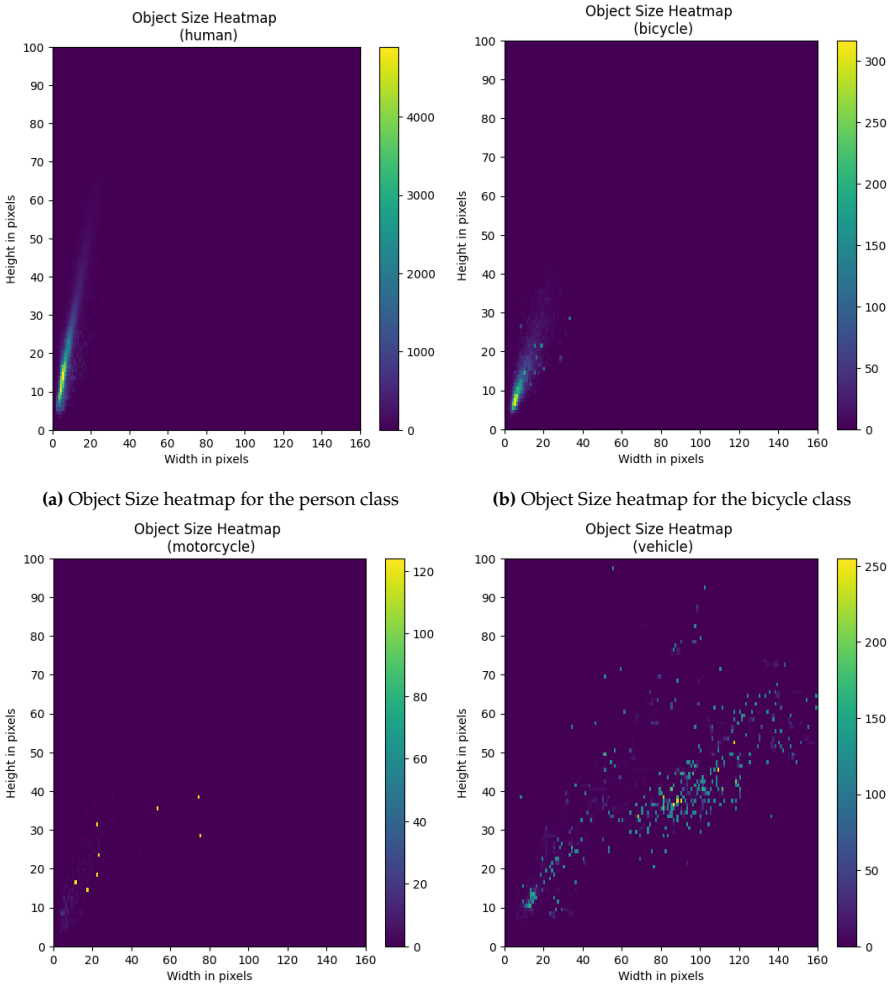
(a) Object Size heatmap for the person class

(b) Object Size heatmap for the bicycle class

(c) Object Size heatmap for the motorcycle class

(d) Object Size heatmap for the vehicle class

Fig. 14: Histograms of training datasplit



(a) Object Size heatmap for the person class

(b) Object Size heatmap for the bicycle class

(c) Object Size heatmap for the motorcycle class

(d) Object Size heatmap for the vehicle class

Fig. 15: Histograms of Valid datasplit

References

[1] C. H. Bahnsen and T. B. Moeslund, "Rain removal in traffic surveillance: Does it matter?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2802–2819, 2018.

[2] H. Bhandari, S. Palit, S. Chowdhury, and P. Dey, "Can a camera tell the weather?" in *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2021, pp. 1–6.

[3] D. Bhattacharjee, T. Zhang, S. Süsstrunk, and M. Salzmann, "Mult: an end-to-end multitask learning transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 031–12 041.

[4] 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: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 2020, pp. 213–229.

[5] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, "Robust video content alignment and compensation for rain removal in a cnn framework," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6286–6295.

[6] W.-T. Chu, X.-Y. Zheng, and D.-S. Ding, "Camera as weather sensor: Estimating weather information from single images," *Journal of Visual Communication and Image Representation*, vol. 46, pp. 233–249, 2017.

[7] K. Dahmane, P. Duthon, F. Bernardin, M. Colomb, F. Chausse, and C. Blanc, "Weathereye-proposal of an algorithm able to classify weather conditions from traffic camera images," *Atmosphere*, vol. 12, no. 6, p. 717, 2021.

[8] R. Dai, M. Lefort, F. Armetta, M. Guillermin, and S. Duffner, "Self-supervised continual learning for object recognition in image sequences," in *Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28*. Springer, 2021, pp. 239–247.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[11] ——, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[13] J. Gao, J. Wang, S. Dai, L.-J. Li, and R. Nevatia, "Note-rcnn: Noise tolerant ensemble rcnn for semi-supervised object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9508–9517.

[14] S. Ghosh, F. Delle Fave, and J. Yedidia, "Assumed density filtering methods for learning bayesian neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[15] D. Glasner, P. Fua, T. Zickler, and L. Zelnik-Manor, "Hot or not: Exploring correlations between appearance and temperature," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3997–4005.

[16] J. C. V. Guerra, Z. Khanam, S. Ehsan, R. Stolkin, and K. McDonald-Maier, "Weather classification: A new multi-class dataset, data augmentation approach and comprehensive evaluations of convolutional neural networks," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 305–310.

[17] S. S. Halder, J.-F. Lalonde, and R. d. Charette, "Physics-based rendering for improving robustness to rain," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 10 203–10 212.

[18] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 15 908–15 919, 2021.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[20] F. Heuer, S. Mantowsky, S. Bukhari, and G. Schneider, "Multitask-centernet (mcn): Efficient and diverse multitask learning using an anchor free approach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 997–1005.

[21] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 1439–1449.

[22] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *CVPR*, 2015.

[23] A. S. Johansen, J. C. J. Junior, K. Nasrollahi, S. Escalera, and T. B. Moeslund, "Chalearn lap seasons in drift challenge: Dataset, design and results," in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Springer, 2022, pp. 755–769.

[24] M. Kieu, A. D. Bagdanov, M. Bertini, and A. Del Bimbo, "Task-conditioned domain adaptation for pedestrian detection in thermal imagery," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 2020, pp. 546–562.

[25] A. Körez, N. Barışçı, A. Çetin, and U. Ergün, "Weighted ensemble object detection with optimized coefficients for remote sensing images," *ISPRS International Journal of Geo-Information*, vol. 9, no. 6, p. 370, 2020.

[26] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, and L. Zhang, "Dn-detr: Accelerate detr training by introducing query denoising," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 619–13 627.

[27] K. Li, Y. Li, S. You, and N. Barnes, "Photo-realistic simulation of road scene for data-driven methods in bad weather," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[28] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, "Single image rain removal via a deep decomposition–composition network," *Computer Vision and Image Understanding*, vol. 186, pp. 48–57, 2019.

[29] D. Lin, C. Lu, H. Huang, and J. Jia, "Rscm: Region selection and concurrency model for multi-class weather recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4154–4167, 2017.

[30] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[31] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[32] I. A. Nikolov, M. P. Philipsen, J. Liu, J. V. Dueholm, A. S. Johansen, K. Nasrollahi, and T. B. Moeslund, "Seasons in drift: A long-term thermal imaging dataset for studying concept drift," in *Thirty-fifth Conference on Neural Information Processing Systems*. Neural Information Processing Systems Foundation, 2021.

[33] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "Spotnet: Self-attention multi-task network for object detection," in *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 230–237.

[34] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: Chances and challenges," in *Proceedings of the 1st international workshop on software engineering for AI in autonomous systems*, 2018, pp. 35–38.

[35] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[36] S. Singh and J. T. Khim, "Optimal binary classification beyond accuracy," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 226–18 240, 2022.

[37] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.

[38] Y. Tian and K. Bai, "End-to-end multitask learning with vision transformer," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[39] M. Tremblay, S. S. Halder, R. De Charette, and J.-F. Lalonde, "Rain rendering for evaluating and improving robustness to bad weather," *International Journal of Computer Vision*, vol. 129, pp. 341–360, 2021.

[40] R. Walambe, A. Marathe, K. Kotecha, G. Ghinea *et al.*, "Lightweight object detection ensemble framework for autonomous vehicles in challenging weather conditions," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[41] H. Wang, Z. Yue, Q. Xie, Q. Zhao, Y. Zheng, and D. Meng, "From rain generation to rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 791–14 801.

[42] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[43] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith *et al.*, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *International Conference on Machine Learning*. PMLR, 2022, pp. 23 965–23 998.

[44] Q. Xiang, L. Zi, X. Cong, and Y. Wang, "Concept drift adaptation methods under the deep learning framework: A literature review," *Applied Sciences*, vol. 13, no. 11, p. 6515, 2023.

[45] R. Ye, B. Yan, and J. Mi, "Bivs: Block image and voting strategy for weather image classification," in *2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET)*. IEEE, 2020, pp. 105–110.

[46] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," *arXiv preprint arXiv:2010.04159*, 2020.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-657-7

AALBORG UNIVERSITY PRESS