Towards Limited Label Learning for Visual Surveillance

Madan, Neelu

DOI (link to publication from Publisher):
10.54337/aau561802540

Publication date:
2023

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

Citation for published version (APA):
Madan, N. (2023). *Towards Limited Label Learning for Visual Surveillance*. Aalborg Universitetsforlag.
https://doi.org/10.54337/aau561802540

TOWARDS LIMITED LABEL LEARNING FOR VISUAL SURVEILLANCE

BY
NEELU MADAN

DISSERTATION SUBMITTED 2023

AALBORG UNIVERSITY
DENMARK

Towards Limited Label Learning for Visual Surveillance

Ph.D. Dissertation

Neelu Madan

Aalborg University
Department of Architecture, Design and Medialogy
Fredrik Bajers Vej 7B
DK-9220 Aalborg

Dissertation submitted: June, 2023

PhD supervisor: Professor Kamal Nasrollahi
Aalborg University

PhD committee: Associate Professor Markus Löchtefeld (chair)
Aalborg University, Denmark

Professor Michael Felsberg
Linköping University, Sweden

Professor Henrik Kastoft
Aarhus University, 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): 978-87-7573-688-1

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

Neelu Madan

Neelu Madan received her B.Sc. degree in Electronics and Communication from Guru Gobind Singh Indraprastha University, Delhi, India. She received her M.Sc. degree in Computer Engineering with a major in Interactive Systems and Visualization from the University of Duisburg-Essen, Germany. She also worked as an intern and software engineer at Robert Bosch GmBH, Germany, and Cadence Design System, India. She started her PhD degree in the Visual Analysis and Perception (VAP) lab, at the Department of Architecture Design and Media Technology in May 2020.

During her PhD, she collaborated with the research group at Milestone Systems A/S on various projects. She is also involved in external collaborations with the Department of Computer Science, University of Bucharest, Romania, and the Center for Research in Computer Vision (CRCV), University of Central Florida, USA. She has been a research scholar for six months at the Center for Research in Computer Vision (CRCV), at the University of Central Florida, USA.

Her main research interests include artificial intelligence, computer vision, machine learning, deep learning, anomaly detection, and self-supervised learning. She has been co-supervising students working in computer vision and deep learning during her PhD.

Curriculum Vitae

Abstract

Visual surveillance is critical for ensuring safety and security, but challenges related to data privacy and annotation biases persist. This thesis proposes solutions to reduce reliance on labeled data in supervised learning for visual surveillance. Initially, it focuses on multi-object tracking and highlights the need for annotated datasets. Subsequently, alternative approaches including synthetic datasets, self-supervised representation learning, and unsupervised learning are explored.

One proposed solution involves synthetic data generation, enabling accurate and automated label generation. The thesis specifically addresses the detection of individuals falling into water by simulating a synthetic thermal dataset. Thermal datasets, with their inherent privacy-preserving nature, are particularly suitable for visual surveillance tasks while complying with data protection regulations.

Furthermore, the thesis addresses generic representation learning through self-supervised tasks on large-scale image datasets unrelated to surveillance. A curriculum learning approach is proposed to acquire robust and transferable representations for limited labeled data in surveillance tasks. Initial results demonstrate improved learned representations for image classification, with potential applications in various visual tasks.

In addition, the thesis presents a solution for unsupervised anomaly detection, eliminating the need for annotated datasets. It introduces self-supervised blocks that incorporate spatial and spatio-temporal contexts through core convolutional layers. These versatile blocks can be integrated into different architectures, resulting in significant performance improvements in anomaly detection. Another solution in the thesis focuses on separate consideration of temporal and spatial cues for anomaly detection. Long-range temporal cues are generated by analyzing social interactions among trajectories, while spatial cues are derived from frame embeddings extracted from video sequences. This integration of cues enhances the system's capability to detect anomalies in complex video data.

Overall, this thesis provides alternative solutions to address bias and noise in labeled datasets for visual surveillance. It emphasizes increasing dataset diversity through synthetic data generation and adopting learning approaches that reduce reliance on labeled data.

Resumé

Video overvågning er afgørende i sikkerhedsøjemed, men der er fortsat udfordringer relateret til datasikkerhed og annoteringsbias. Denne afhandling foreslår løsninger til at reducere afhængigheden af annoterede data til superviseret læring inden for visuel overvågning. Indledningsvis fokuserer afhandlingen på multi-objekt tracking og fremhæver behovet for annoterede datasæt. Efterfølgende udforskes alternative tilgange, herunder syntetiske datasæt, selvsuperviseret repræsentationsindlæring og usuperviseret læring.

En foreslået løsning involverer generering af syntetiske data, hvilket muliggør præcis og automatiseret generering af annoteringer. Afhandlingen behandler specifikt detektering af personer, der falder i vandet ved at simulere et syntetisk termisk datasæt. Termiske datasæt er særligt velegnede til opgaver inden for videoovervågning på grund af deres indbyggede datasikkerhedsegenskaber og overholdelse af databeskyttelsesregler.

Desuden adresserer afhandlingen generisk repræsentationsindlæring gennem selvsuperviserede opgaver på storskala billededatasæt, der ikke er relateret til overvågning. En tilgang til kurriculumindlæring foreslås for at opnå robuste og overførbare repræsentationer med begrænsede mængder annoterede data til overvågningsopgaver. Indledende resultater viser forbedrede lærte repræsentationer til billedklassificering med potentiale for forskellige visuelle opgaver.

Derudover præsenterer afhandlingen en løsning til usuperviseret anomalidetektion, hvilket eliminerer behovet for annoterede datasæt. Den introducerer selvsuperviserede blokke, der inkorporerer rumlig og rumlig-temporal kontekstinformation via kernekonvolutionelle lag. Disse alsidige blokke kan integreres i forskellige arkitekturer og resulterer i betydelige forbedringer i anomalidetektion. En anden løsning i afhandlingen fokuserer på separat overvejelse af tidsmæssige og rumlige indikationer til anomalidetektion. Langsigtede tidsmæssige indikationer genereres ved at analysere sociale interaktioner mellem trajectorier, mens rumlige indikationer opnås gennem billedindlejringer udvundet fra videosekvenser. Denne integration af indikationer forbedrer systemets evne til at opdage anomalier i komplekse videodata.

Samlet set præsenterer denne afhandling alternative løsninger til at håndtere bias og støj i annoterede datasæt til video overvågning. Den lægger vægt på at øge datasættets mangfoldighed gennem generering af syntetiske data, og ved at anvende læringstilgange der reducerer afhængigheden af annoterede data.

Contents

List of publications

The main body of this thesis consists of the following publications:

A **Neelu Madan**, Kamal Nasrollahi, Thomas B Moeslund, "Attention-enabled object detection to improve one-stage tracker," *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys),* vol 2, pp. 736-754, 2022.

B **Neelu Madan**, Mia Sandra Nicole Siemon, Magnus Kaufmann Gjerde, Bastian Starup Petersson, Arijus Grotuzas, Malthe Aaholm Esbensen, Ivan Adriyanov Nikolov, Mark Philip Philipsen, Kamal Nasrollahi, Thomas B Moeslund, "ThermalSynth: A Novel Approach for Generating Synthetic Thermal Human Scenarios", *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, pp. 130-139, 2021.

C Nicolae-Cătălin Ristea, **Neelu Madan**, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, Mubarak Shah, "Self-supervised predictive convolutional attentive block for anomaly detection, "*Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR-Oral)*, pp. 13576-13586, 2022.

D **Neelu Madan**, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, Mubarak Shah, "Self-Supervised Masked Convolutional Transformer Block for Anomaly Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 2023. (Submitted)

E **Neelu Madan**, Arya Farkhondeh, Kamal Nasrollahi, Sergio Escalera, Thomas B Moeslund, "Temporal cues from socially unacceptable trajectories for anomaly detection," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 2150-2158, 2021.

F **Neelu Madan**, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Thomas B Moeslund, "CM-MAE: Curriculum Masking for learning Masking Autoencoders" *Technical Report*, 2023.

Preface

This thesis is submitted as a collection of papers in partial fulfilment of a PhD study at the Section of Media Technology, Aalborg University, Denmark. It is funded by Milestone System A/s. The scientific work focuses on supervised learning and possible alternatives to mitigate bias and labeled data requirements. The proposed solutions include automatic label generation using synthetic datasets and mitigating the requirement for large-scale annotated data using representation learning and unsupervised learning.

This project has been carried out from 2020-2023, mainly in the Visual Analysis and Perception (VAP) Laboratory at Aalborg University, with a research stay at the Center for Research in Computer Vision (CRCV) department at the University of Central Florida, USA.

I would like to thank my supervisor Prof. Kamal Nasrollahi for providing excellent guidance, support, and freedom to pursue my research during the entire PhD project. A big thanks to Prof. Thomas B. Moeslund for providing excellent support and a favourable working environment in the VAP lab. I would like to thank Prof. Mubahark Shah for hosting me at the CRCV lab, Prof. Radu Tudor Ionescu for providing thoughtful and detailed guidance during the projects, Prof. Fahad Shabaz Khan for his valuable insights during the projects, and Prof. Sergio Escalera for his valuable feedback and guidance. I would like to thank fellow PhD students and co-workers at Milestones Research Program at Aalborg University (MRPA), VAP lab, CRCV lab, and Milestone Systems for useful discussions and continuous support. A big thanks to my collaborators, especially Catalin Ristea for bringing positive attitude to projects.

Finally, thanks to all my amazing friends, who have always been supportive and encouraging in stressful situations. I would like to express my heartfelt gratitude to my roommate, Signe Xian Ertbirk, for being my first and most valuable source of support, especially during the challenging times of the COVID-19 pandemic. In closing, I would like to thank my wonderful family, my father, my brothers, my sisters-in-law, and my late mother for their unwavering support.

Neelu Madan
Aalborg University, June 18, 2023

Part I

Overview of the Work

Chapter 1

Introduction

Computer vision is a field that aims to replicate visual perception using artificial intelligence models. While supervised learning has been successful in this field for many years, it heavily relies on large datasets with labeled information specifically tailored to solve particular problems. However, visual perception also involves learning from background knowledge, acquired through simple observation [1]. This type of learning is not dependent on any specific task and cannot be expressed solely in terms of labeled data. Although labeled data is valuable, it introduces biases to the system. These biases can arise from various sources, such as noise in the human annotation process or assumptions made by learning algorithms. One such naive assumption is that training and test data come from the same distribution. However, the real world is inherently complex, and the data encountered at test time may not be part of the training data. Utilizing fewer labeled examples is computationally efficient and mitigates biases. In the context of visual surveillance, this thesis contributes to alternative solutions to supervised learning, which traditionally relies on human-labeled data. These solutions encompass synthetic data, representation learning, and unsupervised learning.

Visual surveillance is one of the most promising computer vision applications, alongside industrial inspection and autonomous driving. Visual surveillance involves collection, processing, and interpretation of video data incoming from various imaging devices, including surveillance cameras and drones. These surveillance systems can be deployed in different environments such as public spaces, hospitals, banks, and government premises. Their primary objective is to detect suspicious activities, track movement, and identify individuals to prevent crime. However, despite the significant social benefits they offer, visual surveillance systems are also among the most controversial AI applications. While they can greatly improve public safety and security, they raise significant ethical and legal concerns regarding individual privacy, bias, and accountability. Therefore, it is crucial to develop AI-based surveillance systems that prioritize individual rights and freedoms while simultaneously enhancing

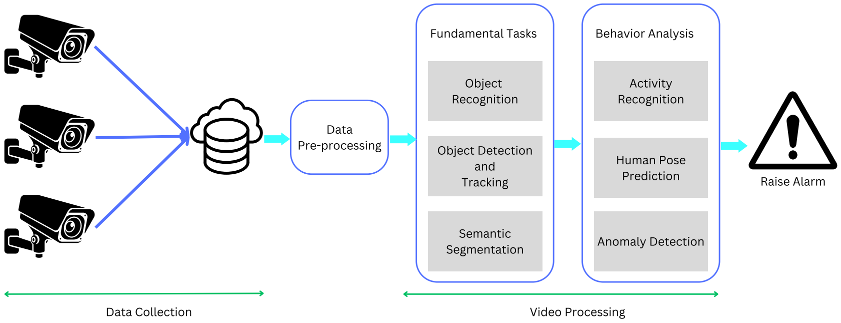


Fig. 1.1: Figure demonstrates a typical surveillance pipeline starting from the data acquisition from video camera to raising an alarm when a suspicious event is detected

public safety. In addressing these issues, this thesis focuses on tackling the problem of fall detection specifically in the thermal domain. The thermal domain is inherently privacy-preserving, aligning with the General Data Protection Regulation (GDPR) [2, 3] guidelines. Additionally, the thesis proposes a representation learning solution that leverages pretraining on ImageNet [4], decoupling the learning process from human-centric factors.

This thesis is part of the Milestone Research Program at Aalborg University (MRPA), which aims to advance research in automatic visual surveillance. Milestone Systems, a company that provides video management systems (VMSs), financed this PhD program. VMSs record and manage video footage of CCTV cameras. The main building blocks of automatic visual surveillance systems are illustrated in Figure 1.1, beginning with the acquisition of video data, followed by data pre-processing. Computer vision techniques, such as object detection, tracking, and recognition, are employed to analyze pre-processed video. Behavioral patterns are then detected, and if potential threats are identified, an automatic alarm is triggered. It is worthwhile to note that visual surveillance systems receive a vast amount of video feed, and not all data is equally valuable.

There are several challenges associated with data acquisition and annotation in automatic visual surveillance. *Firstly*, the data acquisition process is very tedious as it is affected by lighting and weather conditions in outdoor surveillance. As the systems run around the clock, there is a sheer amount of data produced as a result. *Secondly*, even if we obtain large-scale good-quality data, annotating such a huge amount of data is a very expensive process, and is further limited by human mistakes. High-quality labels on the other hand are one of the most critical requirements for an automatic surveillance system as corrupted labels might add noise to the system and thus decreases its robustness. *Third*, the most significant aspect of the surveillance system is privacy concerns. Monitoring of public spaces by surveillance systems can be perceived as an invasion of privacy, as individuals may feel monitored without their ex-

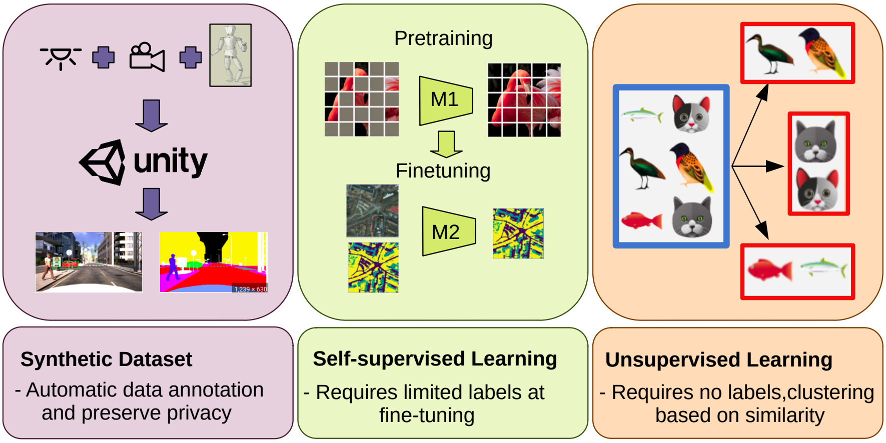

unity
Pretraining
M1
Finetuning
M2
Synthetic Dataset
- Automatic data annotation and preserve privacy
Self-supervised Learning
- Requires limited labels at fine-tuning
Unsupervised Learning
- Requires no labels,clustering based on similarity

Fig. 1.2: Figure demonstrates different approaches discussed in this thesis, based on their reliance on the labeled dataset. Approaches based on synthetic data come with automatic labels, whereas self-supervised and unsupervised require less and no labeled data respectively.

plicit consent. Additionally, surveillance systems can easily be biased towards certain groups and result in discrimination and inequalities in society. These challenges make surveillance dataset distribution even more difficult. This further limits incorporating state-of-the-art methods for automatic visual surveillance. To overcome some of these challenges, this thesis explores alternative solutions to mitigate the reliance on labeled data and introduce a classification problem in the privacy-preserving thermal domain.

This PhD thesis addresses challenges related to data acquisition and labeling in a wide range of visual tasks. Initially, supervised learning is employed to tackle the tracking task, leveraging a lot of human-annotated data. However, dependence on extensive human annotations can add dataset biases in form of noisy labels. Therefore, alternative approaches are explored in this thesis, as also shown in Figures 1.2. One of the alternative solutions involves using synthetic datasets. Although this approach shows promise, its applicability to all computer vision tasks is not universal. To overcome this limitation, the research shifts focus towards more generic solutions, such as self-supervised representation learning and unsupervised learning, which aim to alleviate the data requirement issue. Self-supervised representation learning enables the acquisition of generic representations through self-supervised tasks. This acquired representation can then be fine-tuned to tackle multiple visual tasks, including tracking, object detection, segmentation, and classification, even in scenarios where labeled data is limited. In addition, a substantial part of this thesis addresses the task of unsupervised anomaly detection, which does not depend on the availability of labeled data. Through the exploration and proposition of these alternative solutions, this thesis contributes to the advancement of automatic visual surveillance systems. A short description of each approach in relation to this thesis is described below:

Synthetic datasets (No Human Annotations). Synthetic datasets [5, 6] generated through simulation, offer a valuable resource for training models without the need for human annotations. These datasets provide annotated samples in a cost-effective manner, offering a wide range of diverse data that can contribute to the development of more generalized models. Additionally, synthetic datasets offer privacy-preserving benefits, making them particularly exciting for surveillance applications. However, it's critical to note that they may not generalize well for every visual task as they fail to capture the complexity of the real world.

Self-supervised Representation Learning (Less Labeled Data). Self-supervised representation learning [7–9] is a promising solution for visual surveillance using limited labeled data. By learning a representation from a self-supervised task on a large-scale dataset, model can capture relevant visual features useful for downstream tasks such as tracking and object detection. Once model has learned a reliable representation of data, it can be fine-tuned with limited labels. Furthermore, self-supervised learning can help address the problem of domain shift, where the distribution of the training data is different from that of the test data, by learning more generalizable representations of data.

Unsupervised learning (No Labeled Data). Unsupervised learning [10] is an approach that requires almost no labeled data. In the context of visual surveillance, unsupervised learning can be used for anomaly detection methods [10–13]. Anomaly detection methods can help identify events or activities that do not conform to normal behavior patterns in a given environment. This problem can be approached as a one-class classification problem, where model is trained on normal data and identifies instances that deviate from this normal behavior.

1 Thesis Structure

This PhD thesis primarily contributes to the *Fundamental Tasks* and *behavior analysis* components of the automatic visual surveillance pipeline, as depicted in Figure 1.1. In *Fundamental Tasks*, detection (for tracking) and classification approaches are considered, and in *behavior analysis*, anomaly detection is undertaken. Various computer vision tasks within the context of visual surveillance discussed in this thesis are organized based on their reliance on annotated datasets. Consequently, the thesis is structured into four main research areas: (a) Supervised learning for multi-object tracking, (b) Supervised learning with synthetic datasets for classification, (c) Self-supervised representation learning for generic visual tasks, and (d) Unsupervised learning for anomaly detection.

Figure 1.3 displays publications associated with each research direction. Chapter 2 introduces a supervised approach for learning task-specific representations, targeting multi-object tracking (MOT). This chapter provides the problem statement for this thesis, where large-scale human-annotated datasets are used that account for small improvements in accuracy. Chapter 3 focuses on the generation of a synthetic dataset in the thermal domain. This chapter simultaneously tackles automatic data labeling

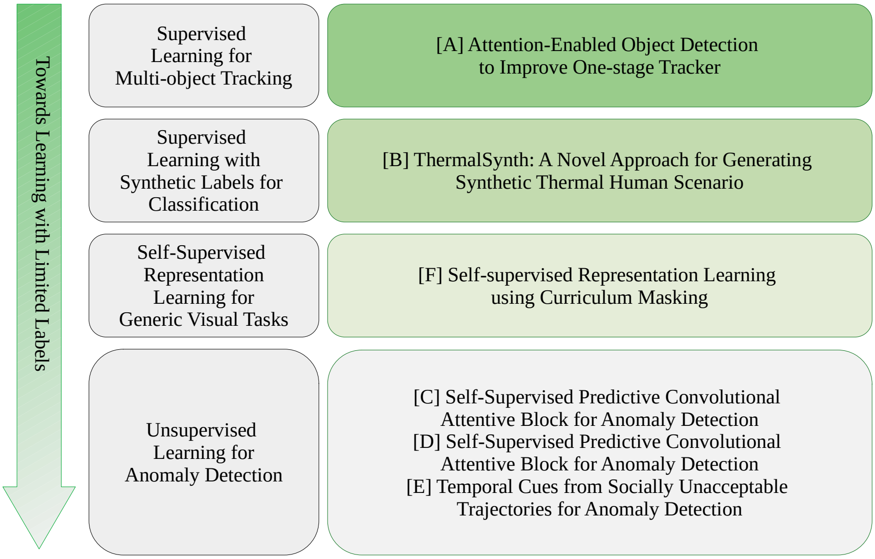

Towards Learning with Limited Labels
Supervised Learning for Multi-object Tracking
[A] Attention-Enabled Object Detection to Improve One-stage Tracker
Supervised Learning with Synthetic Labels for Classification
[B] ThermalSynth: A Novel Approach for Generating Synthetic Thermal Human Scenario
Self-Supervised Representation Learning for Generic Visual Tasks
[F] Self-supervised Representation Learning using Curriculum Masking
Unsupervised Learning for Anomaly Detection
[C] Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection
[D] Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection
[E] Temporal Cues from Socially Unacceptable Trajectories for Anomaly Detection

Fig. 1.3: Figure shows the collection of papers presented in this PhD thesis. Different alternative techniques are arranged in order from requiring full-scale annotated data to no requirement of annotation.

and privacy preservation in surveillance videos.

Chapter 4 delves into generic representation learning by employing a self-supervised task. These representations are usually fine-tuned with limited labeled data to effectively address a wide range of visual tasks. Chapter 5 focuses on providing solutions for anomaly detection in both still images and videos. It introduces a self-supervised task (reconstruction) tailored specifically to anomaly detection, eliminating dependence on labeled data. Finally, chapter 6 concludes this thesis and provides future research directions associated with this project. The key findings in this thesis are listed below:

- The synthetic dataset demonstrates remarkable generalization capabilities, particularly in the context of detecting humans falling into water as proposed in this thesis. Nevertheless, it is important to acknowledge that synthetic datasets may not consistently replicate real-world scenarios. Furthermore, personal bias within synthetic datasets, originating from their creators, adds another layer of complexity to model generalization. In such cases, exploration of solutions independent of labeled data proves valuable.

- Masked image modeling is a powerful pretext task for representation learning. Employing this task enhances representation quality. Preliminary results show an improvement in representation through use of MIM in our proposed curriculum learning framework. The pretraining step requires significant GPU memory and large datasets. However, the acquired representation is applicable to multiple visual tasks.

- Anomaly detection is a crucial problem in visual surveillance. The proposed approaches in this thesis learn specialized representations using reconstruction as a self-supervised task while completely eliminating labeled data requirements. The results highlight the effectiveness of the reconstruction task in providing a strong representation for anomaly detection.

In summary, this thesis proposes potential solutions to address *label bias* in human-annotated datasets. Two approaches are suggested: the utilization of synthetic datasets and the adoption of self-supervised tasks. Synthetic datasets offer the advantage of generating automatically labeled, diverse and accurate data. On the other hand, the self-supervised approach tackles label bias by extracting labels directly from the data itself and mitigates the reliance of pre-defined labels.

References

[1] Y. Bengio, Y. Lecun, and G. Hinton, “Deep learning for ai,” *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.

[2] *General Data Protection Regulation (GDPR) – Official Legal Text*, https://gdpr-info.eu/, (Accessed on 03/02/2022).

[3] A. v. d. B. Paul Voigt, *The EU General Data Protection Regulation (GDPR) - A practical guide*. Cham: Springer International Publishing, 2017.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proceedings of CVPR*, 2009, pp. 248–255.

[5] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, “Motsynth: How can synthetic data help pedestrian detection and tracking?” in *Proceedings of ICCV*, 2021, pp. 10 849–10 859.

[6] C. Pramerdorfer, J. Strohmayer, and M. Kampel, “Sdt: A synthetic multi-modal dataset for person detection and pose classification,” in *Proceedings of ICIP*, 2020, pp. 1611–1615.

[7] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the CVPR*, 2022, pp. 16 000–16 009.

[8] Y. Shi, N. Siddharth, P. H. Torr, and A. R. Kosiorek, “Adversarial masking for self-supervised learning,” in *Proceeding of the International Conference on Machine Learning (ICML)*, 2022.

[9] K. Chen, Z. Liu, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, “Mixed autoencoder for self-supervised visual representation learning,” in *Proceedings of the CVPR*, 2023.

[10] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “Anomaly Detection in Video via Self-Supervised and Multi-Task Learning,” in *Proceedings of CVPR*, 2021, pp. 12 742–12 752.

[11] N. Madan, A. Farkhondeh, K. Nasrollahi, S. Escalera, and T. B. Moeslund, “Temporal cues from socially unacceptable trajectories for anomaly detection,” in *Proceedings of the ICCVW*, 2021, pp. 2150–2158.

[12] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video,” in *Proceedings of CVPR*, 2019, pp. 7842–7851.

[13] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, “Old is Gold: Redefining the Adversarially Learned One-Class Classifier Training Paradigm,” in *Proceedings of CVPR*, 2020, pp. 14 183–14 193.

References

Chapter 2

Supervised Learning for Multi-object Tracking

Supervised learning has long been a dominant approach to solving various visual tasks by leveraging annotated datasets to train deep neural networks (DNNs). With access to labeled data, neural networks perform exceptionally. They have set new benchmarks for tasks ranging from multi-object tracking, object detection, segmentation, and image classification. This chapter discusses a noteworthy use case for supervised learning, focusing on multi-object tracking designed for surveillance scenarios.

1 Introduction

Multi-object tracking (MOT) is an extensively researched field in computer vision that holds substantial importance for various applications. MOT plays a pivotal role in visual surveillance. It enables the monitoring and tracking of objects, such as individuals or vehicles, within a scene. By employing MOT techniques, a system can accurately track multiple objects of interest and generate corresponding trajectories. These trajectories serve as a valuable resource for analyzing object behavior, motion patterns and planning future actions. This information is instrumental in enhancing visual surveillance systems' effectiveness.

MOT is a challenging task for several reasons. *Firstly*, presence of multiple objects in the scene can lead to confusion during their associations, resulting in inaccurate predictions. *Secondly*, poorly captured surveillance videos pose additional challenges, such as poor lighting conditions and varying weather conditions. These factors can make object detection and tracking difficult, as objects may be poorly visible or partially occluded. *Thirdly*, presence of incoming and outgoing objects in videos can result in discontinuous tracks, making it challenging to maintain consistent object associations over time. Additionally, associating objects from different frames can be a complex problem. This chapter aims to address challenges associated with object detection and improves the tracker's performance.

Supervised learning for MOT has shown promising results in recent years, with availability of large-scale annotated datasets such as MOTChallenge [1–3], KITTI [4], and UA-DETRAC [5]. These datasets provide diverse and challenging scenarios for training and evaluating tracking algorithms, facilitating advancements in the field. In MOT, a labeled dataset typically consists of video sequences where each object instance is manually annotated with its corresponding bounding box and trajectory. These annotations serve as ground truth references for training the tracking model. Supervised learning typically involves two stages: training and testing. During the training stage, the model is trained using labeled data. Model captures object appearances, motion patterns, and temporal dependencies. The training process aims to learn the discriminative characteristics of tracked objects and their dynamics using learned representations. Once the tracking model is trained, it is evaluated and tested on previously unseen video sequences during the testing stage. The model attempts to accurately track objects based on learned representations and temporal coherence of object trajectories. Evaluation metrics such as Multi-object tracking accuracy (MOTA) [6], Multi-object tracking precision (MOTP) [6], Higher-order tracking accuracy (HOTA) [7], and tracking speed [6] are used to assess the tracking algorithm's performance.

State-of-the-art (SOTA) approaches in multi-object tracking heavily rely on object detectors, which have achieved remarkable success thanks to large-scale annotated datasets like COCO (Common Objects in Context) [8]. The availability of these datasets has significantly influenced research in multi-object tracking, particularly towards tracking-by-detection approaches [9–13]. These approaches involve first detecting and following multiple objects across consecutive frames of a video to generate their trajectories. The tracking-by-detection approaches are further classified as one-stage tracker [14–16] and two-stage tracker [17–21]. Two-stage trackers use different models for detection and tracking whereas a one-stage tracker uses multi-task learning to perform detection and tracking in a single network. These approaches are discussed in detail, later in this chapter.

The detection step in the tracking-by-detection algorithm is a critical component that directly impacts the accuracy and robustness of the tracking system. Missing detection might result in discontinuous tracking and association problems later on. This chapter, therefore, proposes a solution to improve object detection by incorporating attention modules into one-stage tracking-by-detection algorithms and thus providing a better representation of the objects to be tracked. Attention mechanisms [22, 23], such as spatial [22, 23] and channel attention [22, 23], are applied to the feature maps generated by the detection network to emphasize the most informative regions and channels, thereby enhancing detection performance.

2 State-of-the-Art

Multi-object Tracking. The development of novel algorithms and methods has led to significant advancements in multi-object tracking over the years. A remarkable improvement in performance has been achieved in the field, owing to deep learning methods like convolutional neural networks (CNNs) [24] and transformer models [25]. Trackers combine several cues, including motion cues, appearance cues, and location cues, to ensure robust tracking. The initial MOT approaches use probabilistic models such as the Kalman filter [26, 27], Bayesian filtering [28], and particle filter [29] to capture motion cues; and handcrafted features such as color histograms [30], edge detectors [31], and corner detectors [31] to obtain appearance cues. Finally, Hungarian matching [32] is performed to associate the objects represented by these joint cues (motion and appearance). Later on, deep learning-based approaches such as recurrent neural networks (RNNs) [33] for motion cues, and Re-Identification (Re-IDs) based on CNNs for appearance cues replaced traditional approaches. The association task in these networks is also accomplished by deep models such as DeepSORT [20]. The most recent trackers however utilize transformer models [34–36], which learn all the necessary cues via self- and cross-attention among the video frames in a single model.

SOTA tracking approaches can be classified into multiple categories based on different criteria. One such taxonomy is based on the initialization method, where tracking algorithms can be classified as detection-based or detection-free. Detection-based methods [9–12] rely on pre-trained object detectors and link detected objects in each frame to generate trajectories. In contrast, detection-free methods [37–40] require manual initialization of objects in the first frame, followed by localizing those objects in the subsequent frames. Another taxonomy is based on processing mode, where tracking methods can be classified as online or offline. Online tracking [37–43] only depends on past frames to decide the association of objects in the current frames, while offline tracking [44–49] depends on both past and future frames to decide the associations in the current frame. These approaches differ in their strengths and weaknesses. They may be most appropriate for specific applications depending on computing efficiency, accuracy, and complexity of the scene. When it comes to visual surveillance, online detection-based trackers are suitable since they can only make decisions based on past frames. Moreover, detection-based tracking being independent of initialization protocols provides flexibility for objects incoming and leaving the frames and thus proves to be a suitable choice for surveillance scenarios.

The most successful methods for MOT follow the tracking-by-detection approach [9, 10, 16, 50], detecting objects in each frame and then associating them over time. The existing approaches in this research area are classified into two major groups namely: one-stage trackers [14–16], and two-stage trackers [17–21]. The one-stage trackers use a single model for detection and association whereas the two-stage tracker employs two separate models for detection and association tasks. The speed of two-stage trackers however remains a problem, as they are computationally expensive and

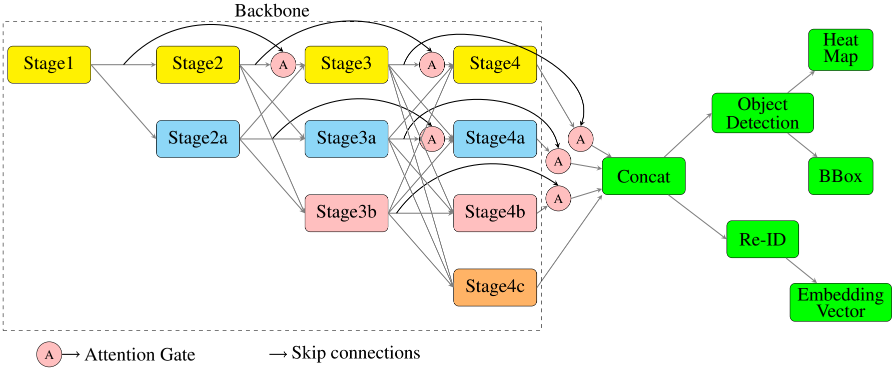


Fig. 2.1: Implicit attention, where attention gates are used to pass the features across the consecutive stages without using any additional loss function, the backbone architecture, in this case, is based on HRNet [53] [This figure is taken from [13]].

often slow in inference, making them unsuitable for real-time applications like visual surveillance. To address this, one-stage trackers have emerged, benefiting from multi-task learning [51, 52] that enables them to train a joint network for both detection and association. This reduces the computational complexity of the network, allowing for real-time inference, and making one-stage trackers increasingly popular for real-time object-tracking applications.

This chapter introduces the one-stage tracker approach, which is a requirement for visual surveillance due to its fast inference time. The proposed idea incorporates attention into these trackers. This attention-based detection technique can be used to detect objects in challenging scenarios such as low-light and highly cluttered environments.

3 Scientific Contributions

This section discusses the approach used in Paper A to track multiple objects using large human-annotated datasets.

The approach presented in Paper A is inspired by tracking-by-detection, particularly one-stage trackers with low computational complexity. These trackers typically utilize object detection backbones and employ multi-task learning to perform both detection and association within the same framework. The proposed approach incorporates HRNet [53] as a backbone to facilitate multi-task learning. HRNet [53] is a well-known architecture for object detection that employs parallel convolutions, with each branch representing a different resolution (refer to Figures 2.1 and 2.2). The method incorporates attention into the one-stage tracking framework using attention gates. The paper explores two approaches to incorporate attention: implicit and explicit attention.

The proposed implicit attention method incorporates skip connections, inspired by

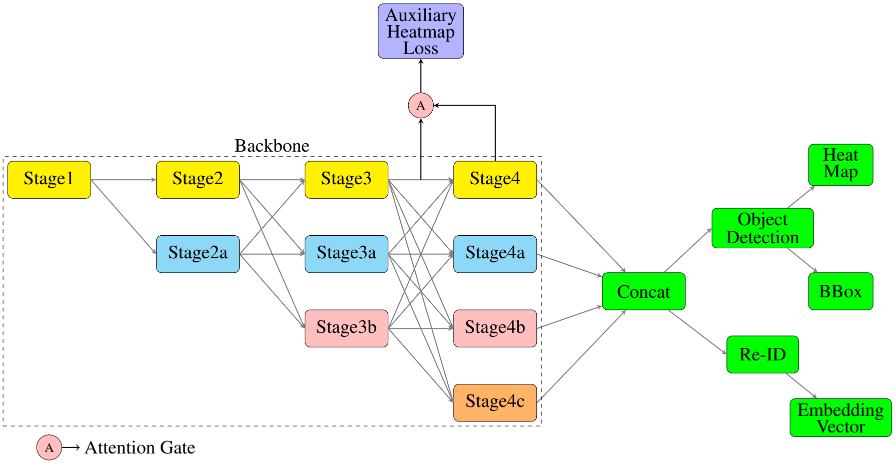


Fig. 2.2: Explicit attention, where attention gates are passing the feature at the highest resolution between the last two consecutive stages, where the attention gates are guided via auxiliary heat-map loss. The backbone architecture, in this case, is based on HRNet [53]. [This figure is taken from [13]].

the ResNet architecture [54], to improve feature propagation among different stages. The paper empirically shows that incorporating skip connections between consecutive stages and adding attention gates for each connection improves the results. The network architecture of the proposed one-stage tracker based on HRNet [53] with implicit attention is illustrated in Figure 2.1.

Explicit attention introduces gates between the last two stages at the highest resolution of HRNet [53]. Furthermore, in the case of explicit attention, Paper A introduces an auxiliary loss function that minimizes heatmap loss. The results indicate that explicit attention further improves multi-object tracking performance. The network architecture of the proposed one-stage tracker based on HRNet [53] with explicit attention is shown in Figure 2.2.

The quantitative results of incorporating both implicit and explicit attention mechanisms are presented in Table 2.1. The results show that both approaches outperform the state-of-the-art (SOTA) methods on MOT16 [1] and MOT17 [1] benchmarks in terms of MOTP and IDF1 metrics. Moreover, the explicit attention-based approach surpasses the implicit attention-based approach, indicating that the introduction of an auxiliary loss function enhances feature propagation across stages. Paper A also includes a comprehensive ablation study that investigates the rationale behind incorporating skip connections between consecutive stages in implicit attention. The paper also includes a study demonstrating empirically that introducing the guiding function at stage 4, corresponding to the highest resolution, leads to improved performance. Overall, the results show that improved object detection can lead to enhanced object tracking. We infer the reason for the improvement is that missing object detections in some frames result in incorrect associations.

Tracker	Publication	Year	MOT16				MOT17			
			MOTA↑	MOTP↑	IDF1↑	IDs↓	MOTA↑	MOTP↑	IDF1↑	IDs↓
DeepSORT [20]	ICIP	2017	61.4	79.1	62.2	**781**	60.3	79.1	61.2	**2442**
Tracktor+CTdet [11]	ICCV	2019	-	-	-	-	54.4	78.1	56.1	2574
JDE [14]	ICCV	2019	64.4	55.8	1881	-	-	-		
CenterNet [50]	ECCV	2020	-	-	-	-	**67.8**	-	64.7	3039
Chained-Tracker [10]	ECCV	2020	**67.6**	78.4	57.2	1897	66.6	78.2	57.4	5529
Ours(Implicit)	IntelliSys	2021	64.6	78.8	65.9	1234	63.2	78.7	64.8	3357
Ours(Explicit)	IntelliSys	2021	64.9	**79.7**	**66.4**	1489	63.7	**79.1**	**66.0**	3696

Table 2.1: Comparing the proposed systems against SoTA two-stage and one-stage trackers that have reported their results on MOT16 and MOT17 [This table and caption is taken from [13]].

4 Summary

The chapter discusses an approach to multi-object tracking following tracking-by-detection approaches. As surveillance applications need to be real-time, one-stage trackers in this category that perform the joint learning of object detection and association are used in this chapter. The main contribution made through this work is highlighted below:

- The proposed implicit attention propagates features across consecutive stages of an HRNet [53] via attention gates. Incorporating implicit attention into HRNet [53] leads to improvements over the SOTA on MOTP [6] and IDF1 [6] metrics.
- The proposed explicit attention utilizes skip connections with attention gates at the highest resolution and incorporates an auxiliary heat-map loss in the HRNet [53]. Incorporating explicit attention into HRNet [53] results in improvements over the proposed implicit attention and subsequently the SOTA on MOTP [6] and IDF1 [6] metrics

This research work benefited from the conventional supervised learning approach discussed earlier in this chapter. The labeled datasets used for training the proposed system are: CityPersons (CP) dataset [55], CalTech (CT) dataset [56], MOT16 [1], MOT17 [1], CUHK-SYSU (CS) dataset [9], and PRW dataset [57]. These datasets include scenes with varying crowd densities. Generating such datasets through manual annotation would be an intensive and laborious task. Furthermore, manual annotations are prone to error, which introduces noise during training and consequently introduce bias. This thesis discusses some solutions to reduce the requirement of manually annotated datasets. The next chapter in this thesis proposes a solution using a synthetic dataset, where annotations are generated automatically.

References

[1] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “Mot16: A benchmark for multi-object tracking,” *arXiv*, 2016.

[2] P. Dendorfer, H. Rezatofighi, A. Milan, J. Q. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, and L. Leal-Taix'e, “Mot20: A benchmark for multi object tracking in crowded scenes,” *ArXiv*, 2020.

[3] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. D. Reid, S. Roth, and L. Leal-Taixé, “Motchallenge: A benchmark for single-camera multiple target tracking,” *International Journal of Computer Vision (IJCV)*, vol. 129, pp. 845 – 881, 2020.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, pp. 1231 – 1237, 2013.

[5] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, “Detrac: A new benchmark and protocol for multi-object tracking,” *ArXiv*, 2015.

[6] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

[7] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, “Hota: A higher order metric for evaluating multi-object tracking,” *International journal of computer vision*, vol. 129, pp. 548–578, 2021.

[8] 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 *Proceeding of the ECCV*, 2014, pp. 740–755.

[9] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint detection and identification feature learning for person search,” in *Proceedings of the CVPR*, 2017.

[10] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” in *Proceedings of the ECCV*, 2020.

[11] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, “Tracking without bells and whistles,” in *Proceedings of the ICCV*, 2019, pp. 941–951.

[12] R. Li, B. Zhang, J. Liu, W. Liu, and Z. Teng, “Inference-domain network evolution: A new perspective for one-shot multi-object tracking,” *IEEE Transactions on Image Processing*, vol. 32, pp. 2147–2159, 2023.

[13] N. Madan, K. Nasrollahi, and T. B. Moeslund, “Attention-enabled object detection to improve one-stage tracker,” in *Proceedings of the Intelligent Systems Conference (IntelliSys)*, 2022, pp. 736–754.

[14] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, “Towards real-time multi-object tracking,” in *Proceedings of the ECCV*, 2020, p. 107–122.

[15] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, “Mots: Multi-object tracking and segmentation,” in *Proceedings of the CVPR*, 2019.

[16] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “Fairmot: On the fairness of detection and re-identification in multiple object tracking,” *International Journal of Computer Vision*, vol. 129, pp. 3069–3087, 2021.

[17] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, “Poi: Multiple object tracking with high-performance detection and appearance feature,” in *Proceedings of the ECCV Workshops*, 2016.

[18] X. Wan, J. Wang, Z. Kong, Q. Zhao, and S. Deng, “Multi-object tracking using online metric learning with long short-term memory,” in *Proceeding of the IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 788–792.

[19] K. Fang, Y. Xiang, X. Li, and S. Savarese, “Recurrent autoregressive networks for online multi-object tracking,” in *Proceeding of the WACV*, 2018, pp. 466–475.

[20] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proceeding of IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[21] N. Mahmoudi, M. A. Seyed, and M. Rahmati, “Multi-target tracking using cnn-based features: Cnnmtt,” *Multimedia Tools and Applications*, vol. 78, no. 6, pp. 7077–7096, 2019.

[22] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *Proceedings of the ECCV*, 2018, p. 3–19.

[23] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, “Bam: Bottleneck attention module,” in *Proceedings of BMVC*, 2018.

[24] Y. Lecun and Y. Bengio, *Convolutional Networks for Images, Speech and Time Series*, 1995, pp. 255–258.

[25] 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 *Proceedings of the ICLR*, 2021.

[26] “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[27] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[28] S. Srkk, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.

[29] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, “The unscented particle filter,” in *Advances in Neural Information Processing Systems*, vol. 13. MIT Press, 2000.

[30] K. Du, Y. Ju, Y. Jin, G. Li, S. Qian, and Y. Li, “Meanshift tracking algorithm with adaptive block color histogram,” in *Proceedings of Conference on Consumer Electronics, Communications and Networks (CECNet)*, 2012, pp. 2692–2695.

[31] D. Jang and H.-I. Choi, “Moving object tracking using active models,” in *Proceedings of International Conference on Image Processing (ICIP)*, 1998, pp. 648–652 vol.3.

[32] H. W. Kuhn *et al.*, “The hungarian method for the assignment problem,” *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[33] Y. Zhang, Y. Ming, and R. Zhang, “Object detection and tracking based on recurrent neural networks,” in *2Proceeding of IEEE International Conference on Signal Processing (ICSP)*, 2018, pp. 338–343.

[34] P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, and P. Luo, “Transtrack: Multiple-object tracking with transformer,” *arXiv preprint arXiv: 2012.15460*, 2020.

[35] T. Meinhardt, A. Kirillov, L. Leal-Taixé, and C. Feichtenhofer, “Trackformer: Multi-object tracking with transformers,” in *Proceedings of the CVPR*, 2022, pp. 8844–8854.

[36] F. Zeng, B. Dong, Y. Zhang, T. Wang, X. Zhang, and Y. Wei, “Motr: End-to-end multiple-object tracking with transformer,” in *Proceedings of the ECCV*, 2022.

[37] W. Hu, X. Li, W. Luo, X. Zhang, S. Maybank, and Z. Zhang, “Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2420–2440, 2012.

[38] L. Zhang and L. van der Maaten, “Structure preserving object tracking,” in *Proceedings of the CVPR*, 2013, pp. 1838–1845.

[39] ——, “Preserving structure in model-free tracking,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 756–769, 2014.

[40] J. Zhang, L. L. Presti, and S. Sclaroff, “Online multi-person tracking by tracker hierarchy,” in *Proceedings of Advanced Video and Signal-Based Surveillance (AVSS)*, 2012, pp. 379–385.

[41] C. Wu, H. Sun, H. Wang, K. Fu, G. Xu, W. Zhang, and X. Sun, “Online multi-object tracking via combining discriminative correlation filters with making decision,” *IEEE Access*, vol. 6, pp. 43 499–43 512, 2018.

[42] Y. Xiang, A. Alahi, and S. Savarese, “Learning to track: Online multi-object tracking by decision making,” in *Proceedings of the ICCV*, 2015, pp. 4705–4713.

[43] J. H. Yoon, C.-R. Lee, M.-H. Yang, and K.-J. Yoon, “Online multi-object tracking via structural constraint event aggregation,” in *Proceedings of the CVPR*, 2016, pp. 1392–1400.

[44] Z. Qin and C. R. Shelton, “Improving multi-target tracking via social grouping,” in *Proceedings of the CVPR*, 2012, pp. 1972–1978.

[45] B. Yang and R. Nevatia, “An online learned crf model for multi-target tracking,” in *Proceedings of the CVPR*, 2012, pp. 2034–2041.

[46] C.-H. Kuo, C. Huang, and R. Nevatia, “Multi-target tracking by on-line learned discriminative appearance models,” in *Proceedings of the CVPR*, 2010, pp. 685–692.

[47] J. F. Henriques, R. Caseiro, and J. Batista, “Globally optimal solution to multi-object tracking with merged measurements,” in *Proceedings of the ICCV*, 2011, pp. 2470–2477.

[48] D. Sugimura, K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, “Using individuality to track individuals: Clustering individual trajectories in crowds using local appearance and frequency trait,” in *Proceedings of the ICCV*, 2009, pp. 1467–1474.

[49] W. Choi and S. Savarese, “Multiple target tracking in world coordinate with single, minimally calibrated camera,” in *Proceedings of the ECCV*, 2010.

[50] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” in *Proceedings of the ECCV*, 2020.

[51] R. Cipolla, Y. Gal, and A. Kendall, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *Proceedings of the CVPR*, 2018, pp. 7482–7491.

[52] I. Kokkinos, “Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory,” in *Proceedings of the CVPR*, 2017, pp. 5454–5463.

[53] Y. Li, C. Wang, Y. Cao, B. Liu, Y. Luo, and H. Zhang, “A-hrnet: Attention based high resolution network for human pose estimation,” in *Proceedings of International Conference on Transdisciplinary AI (TransAI)*, 2020, pp. 75–79.

[54] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proceedings of CVPR*, 2016, pp. 770–778.

[55] S. Zhang, R. Benenson, and B. Schiele, “Citypersons: A diverse dataset for pedestrian detection,” in *Proceedings of the CVPR*, 2017.

[56] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” in *Proceedings of the CVPR*, 2009.

[57] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, “Person re-identification in the wild,” in *Proceedings of the CVPR*, 2017, pp. 1367–1376.

Chapter 3

Supervised Learning with Synthetic Datasets for Classification

Synthetic datasets have demonstrated success in various visual tasks [1–5], highlighting their potential advantages. This thesis primarily focuses on surveillance applications, wherein one prominent benefit of synthetic datasets is their ease of distribution compared to real datasets. Synthetic datasets mitigate privacy concerns and eliminate personal identity exposure. However, to ensure the efficacy of synthetic datasets, it is imperative to develop robust domain adaptation methodologies. Thoroughly understanding the problem definition is essential, as it allows identifying specific areas where synthetic datasets can offer significant advantages over real datasets. This chapter explores the fall classification problem as a specific case, emphasizing the effective utilization of synthetic datasets to address it.

1 Introduction

In recent years, researchers have been using synthetic datasets [2–7] with freely available annotations as a potential solution to the data-annotation problem in supervised learning. Synthetic datasets offer several benefits for computer vision tasks. *Firstly*, they provide a cost-effective alternative to manual annotation, enabling the rapid generation of large volumes of labeled data at a lower cost. *Secondly*, synthetic datasets allow control over various factors, including lighting conditions, camera perspectives, and object variations. This facilitates the creation of diverse and well-controlled datasets that train models to be robust in different real-world scenarios. *Thirdly*, synthetic datasets inherently possess ground truth annotations since the virtual environment allows for precise labeling of objects and their attributes. These readily available annotations eliminate manual annotation efforts. *Lastly*, synthetic datasets can be easily shared, replicated, and distributed among researchers, thereby promoting reproducibility and facilitating collaboration within the computer vision community.

Despite their advantages, synthetic datasets have certain limitations that should be considered during generation. *Firstly*, simulated examples may not fully capture real world complexities and nuances. *Secondly*, existing simulators may struggle to accurately replicate domain-specific features, making the simulation of certain datasets, such as medical datasets, challenging compared to natural images. *Thirdly*, these datasets are often generated based on assumptions, which can introduce personal biases that propagate through the model training process. *Additionally*, synthetic datasets do not always ensure seamless domain adaptation from the synthetic environment to the complex real world. As a result, a combination of real and synthetic datasets is sometimes used to train neural networks to capture finer details.

Nonetheless, synthetic datasets have trained deep neural networks in recent years. Multiple use cases have demonstrated that training neural networks with synthetic datasets results in generalized and robust models. For example, Fabbri *et al.* [6] demonstrated the superior performance of networks trained on synthetic datasets for object detection and multi-object tracking, while Ros *et al.* [5] showcased the effectiveness of generating large-scale segmentation datasets for training robust models. This chapter focuses on a special approach where synthetic datasets are created by combining real-world backgrounds with simulated foregrounds. The generated dataset is then used to solve a classification problem of identifying people falling into water. The results indicate that training networks on synthetic datasets exhibit exceptional domain transfer capabilities in the real world for this specific application.

Object classification is a well-researched area in computer vision, assigning predefined labels or categories to objects observed in images or videos. This task is crucial for image understanding, object recognition, and scene comprehension. Large-scale annotated datasets like ImageNet [8] have played a pivotal role in advancing research in this domain. However, this chapter focuses on a more specific and simplified two-way classification problem: identifying instances of individuals falling into bodies of water. Identifying such instances poses unique challenges due to data scarcity and risks. Individuals falling into water bodies are relatively rare, making it challenging to gather a sufficiently large and diverse dataset for training deep neural networks. To overcome this limitation, this chapter proposes a methodology that utilizes synthetic datasets generated in the privacy-preserving thermal domain.

The proposed synthetic dataset shows promising results in terms of its ability to generalize to real-world scenarios of individuals falling into water. The evaluation involves real-fall scenarios where individuals voluntarily jump into the water, ensuring authentic and representative data samples.

2 State-of-the-Art

Towards Synthetic Datasets. Generating synthetic datasets refers to a process of creating artificial data that mimics real-world data. Synthetic data can supplement or replace real data in machine learning tasks. This is particularly important when real data is limited or when labeling real data costs high. Moreover, the introduction of data protection regulations like the European general data protection regulation (GDPR) [9, 10], enforces strict regulations to retain the identities of the test subjects without information which further limits the scope to collect real data in some cases. The deep neural networks however are data-hungry and the challenges associated with data acquisition limit the models' ability to perform well on visual tasks. As a result, synthetic datasets [6, 7, 11–15] provide a suitable alternative, especially for applications such as surveillance, which involves critical information like humans as test subjects. Synthetic data can be created using various techniques such as data augmentation [11–13], generative models [15], and simulation [6, 7].

During the learning process of computer vision tasks such as image classification, object detection, and segmentation, a variety of data-augmentation techniques such as random flipping [16], cropping [16], and scaling [16] are commonly employed. These augmentations incorporate diversity in the data and prevent network overfitting. However, there are some approaches that have started using these augmentation techniques to manipulate training datasets specific to problem domains. One such problem domain is unsupervised image/video anomaly detection, where only normal samples are available at training time. Some approaches [13, 17, 18], therefore, generate fake anomalies using data-augmentation techniques like cut-paste [18], random-cutouts [17], and skipping frames in videos [13]. This results in performance improvement for anomaly detection by introducing these fake anomalies, generated by data-augmentation techniques.

Unlike data-augmentation methods, where we apply different manipulations to training samples to increase the sample size. The generative models such as generative adversarial networks (GANs) [19], and diffusion models [20] learn to synthesize fake samples to increase the training sample size. Kniaz *et al.* [15] uses a style-GAN [14] approach to synthesize the RGB images to the thermal domain due to the scarcity of labeled datasets in the thermal domain. Some recent approaches [21–24] are learning image manipulation and generating fake images based on the text prompts.

Generating a synthetic dataset [6, 7, 25, 26] based on simulation first requires setting up a simulation environment that resembles real-world characteristics such as lightning variation, possible motion characteristics, and object appearance. These simulations are designed on 3D environment simulators such as Unity [27], Blender [28], VIVID [29], Habitat 2.0 [30], and Unreal [31]. After setting up simulation parameters in a simulator, a synthetic dataset is created by running these simulations. The generated synthetic dataset can now be used for solving different visual tasks. One of the major benefits of generating synthetic datasets in this way is that accurate annotations can be generated automatically. This mitigates the problem of manual annotations of large datasets. A few successful attempts are being made in the RGB domain [2, 3, 5, 6, 32] to generate photo-realistic synthetic data. These datasets illustrate exceptional transferability characteristics to the real-world dataset for visual tasks like multi-object tracking and segmentation.

However, the generation of thermal datasets synthetically using 3D virtual environments is a relatively limited practice, as evidenced by only a few instances [6, 7, 25, 26]. For example, Pramerdorfer et al. [7] employ Blender [33] to generate synthetic depth and thermal images depicting human behavior in indoor environments. Similarly, Blythman et al. [25] utilize Zephyr [34] to generate synthetic thermal representations of human heads placed in cars. Another approach by Bongini et al. [26] involves using Unity [35] to generate synthetic thermal videos by combining 3D foreground objects with real background images in autonomous driving scenarios. This chapter discusses an approach to generate synthetic humans in unity [35] and combines it with the real background from the long-term drift (LTD) dataset [36] to produce a diverse dataset of humans falling into water.

Fall Classification. The detection of people falling in various scenarios carries significant implications, particularly for vulnerable populations such as the elderly or individuals with specific medical conditions. Falling incidents can result in serious consequences, and the severity may vary depending on the surrounding context, especially in outdoor environments. Hence, addressing the fall classification problem is essential for societal success. Visual surveillance systems offer potential solutions for monitoring such occurrences. However, the development of a highly accurate machine learning model is imperative to ensure minimal undetected cases. This will enable the automatic detection and reporting of life-threatening and critical issues. This subsection discusses some of the previous methods that aim to automate the process for both indoor and outdoor fall cases.

A considerable amount of research [37–44] has been dedicated to addressing the issue of falls among the elderly and individuals with health conditions in indoor environments. Various modalities have been explored in this research area, including wearable sensors [45], acoustics [46], pressure-based systems [45], and artificial vision techniques [42]. Some early approaches based on computer vision for fall detection [42–44] relied on manually engineered features, such as aspect ratios of bounding boxes, combined with classifiers like support vector machines (SVMs) [47]. However, these methods have gradually been superseded by convolutional neural networks (CNNs) [37–41].

Limited attention is currently given to fall incidents occurring in outdoor settings, such as individuals falling into water [48–50] or on streets [51, 52]. Despite the existence of qualitative studies [51, 52] that focus on falls in outdoor settings such as sidewalks, streets, and garages, effectively capturing such events through surveillance cameras remains a challenging task. This difficulty stems partly from the scarcity of available samples, particularly for instances of people falling into the water. Among the first attempts to tackle this issue, Bonderup *et al.* [48] proposes a thermal-based pipeline encompassing person detection, person tracking, fall prediction, and fall detection. To augment their approach, Bonderup *et al.* [48] generates a thermal fall dataset by prompting individuals to intentionally jump into the water. Subsequently, Nikolov *et al.* [50] presents a dataset for human falls, which involves simulating scenarios where a dummy is thrown into the water. The authors employ optical flow maps within a designated region of interest to detect falls. This chapter presents a methodology for generating a synthetic dataset in the thermal domain. It focuses on scenarios involving individuals falling into a harbor.

3 Scientific Contributions

This section provides a comprehensive overview of an approach that generates a synthetic dataset specifically tailored to people falling into water. The problem at hand is of great significance, as it involves rare and life-threatening events where individuals encounter water hazards. Collecting real-world datasets for such events poses significant challenges, making synthetic datasets a valuable solution in this case. This section presents an approach documented in Paper B of this thesis.

Paper B proposes a method for generating a synthetic dataset through simulation in the thermal domain. The annotations for this dataset are generated automatically with a single click using the Unity perception package [27]. The GDPR rules [9], which prioritize data privacy, have highlighted the value of thermal domain images, as they preserve privacy. Paper B proposes a novel approach by combining synthetic thermal foreground objects with real backgrounds. The synthetic foreground objects in the thermal domain are generated using the Stefan-Boltzmann law [53], which calculates the black body radiation (j^*) of an object as shown in Equation 3.1. In the equation, T represents the absolute temperature of the object in Kelvins, ϵ_m denotes the thermal emissivity of the material, and σ is the Stefan-Boltzmann constant [54], equal to $5.6704 \times 10^{-8} \frac{W}{m^2 K^4}$. To simplify the data generation process, a constant value of $T = 300.4$ is used. The emissivity values for human skin and cotton are taken from Table 3.1. The synthetic foreground objects in this dataset are currently limited to humans, so only human skin and cotton clothing materials are considered in this case.

$$j^* = \epsilon_m \; \sigma \; T^4 \tag{3.1}$$

The final pixel value (p) for thermal objects is computed using Eq. 3.2, where $\mathcal{L}$ and G represent level and gain respectively. Paper empirically verifies that choosing the value of $\mathcal{L} = 20$ and $G = 0.05$ provides excellent results in terms of photo-realistic thermal object appearance.

$$p = (j^* \cdot G) + \mathcal{L} \tag{3.2}$$

Furthermore, material emissivity ϵ_m is combined with the color emissivity ϵ_c with a blending factor of 0.31 and used in Eq. 3.1. ϵ_c is computed via Eq. 3.3, where L stand for Luminance and calculated by combining R,G, and B channels as $L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B$. Luminance represents the albedo texture of each pixel.

$$\epsilon_c = (1 - L) \cdot 0.15 + 0.84 \tag{3.3}$$

The proposed approach in this paper mostly generates synthetic humans as foreground objects in the thermal domain. These humans are combined with the real background obtained from the LTD dataset [36]. This combination allows the generated dataset to capture all the seasonal variations present in the LTD dataset [36]. The paper proposes simulating diverse fall scenarios to generate the fall dataset.

We generate 310,788 synthetic thermal image frames, comprising an equal number of fall and non-fall frames. To incorporate temporal information, we combine three consecutive frames as a 3-channel image. This results in a synthetic dataset of 103,596 images with equal numbers of fall and non-fall sequences. The dataset includes automatically generated foreground object segmentations, which are not utilized for the classification problem addressed in this paper. However, these annotations label the samples into fall and non-fall categories. Synthetic data is generated using a static camera. This makes it easier to mask out the harbor region and annotate the dataset based on the location of foreground objects.

The proposed synthetic dataset in the thermal domain aims to address the life-threatening issue of detecting people falling into the harbor. To achieve this, we trained popular classification models such as AlexNet [56] and ResNet [57] on the synthetic dataset. For testing the system, we have two similar thermal datasets. The first dataset consists of instances where people voluntarily jump into the water (referred to as the real dataset), while the second dataset involves people throwing human-shaped dolls into the water to mimic human-falling situations.

We evaluate the models on both datasets and present the classification results as confusion matrices in Figure 3.1. The results demonstrate that the models trained on the synthetic dataset (AlexNet and ResNet) generalize well to the real dataset. However, in this case, the simpler AlexNet model performs better than the ResNet model, accurately detecting almost every fall incident in the real dataset. On the other hand, the synthetic dataset fails to generalize effectively to the dummy dataset due to a significant difference in fall behavior. The dummy humans in this dataset exhibit

Emissivity Values				
Human Skin	Cotton	Asphalt	Water	Snow
0.95	0.95	0.95	0.93	0.90

Table 3.1: Emissivity coefficients of different materials, which can be plugged into Stefan-Bolzmann law to compute the black-body radiation of the object. The data-generation approach in this chapter considered this computation for human skin and Cotton (for cloth material), taken from [53] [Source: Table is taken from [55]]

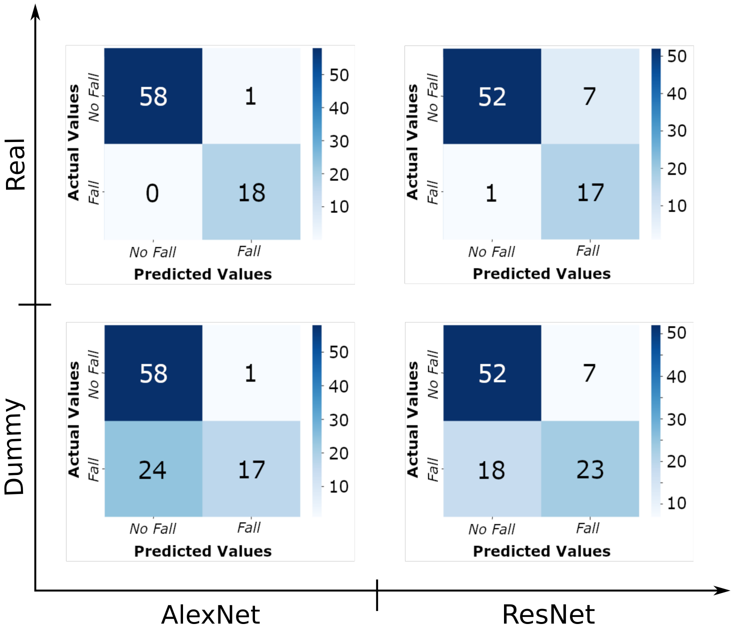


Fig. 3.1: Confusion Matrices for classification on real (top) and dummy (bottom) datasets containing samples of people falling into the water, computed based on two classical classification approaches namely AlexNet(left) and ResNet (right) [Figure is taken from [55]]

a distinct falling tendency by sticking to the pier, which deviates from the typical behavior of human falls.

4 Summary

This chapter introduces a novel approach to tackle data annotation challenges by utilizing a synthetic dataset. The synthetic dataset provides automatically generated labels, which offer high accuracy. The proposed approach focuses on generating the dataset specifically for the privacy-preserving thermal domain, making it well-suited to surveillance applications. The key scientific contributions to this work are as follows:

- The development of a novel solution for generating a synthetic dataset in the thermal domain. This is done by combining real background with synthetic foreground objects. This approach enhances diversity in the generated dataset by including the real backgrounds varying in appearance due to seasonal changes, as discussed by Nikolov *et al.* [36].
- The proposal of a privacy-preserving thermal synthetic dataset for the critical problem of detecting people falling into the water.

- Experimental demonstration of the excellent transferability of the generated thermal synthetic dataset to real-world falling scenarios. This dataset can contribute significantly to societal benefit by enabling the timely detection and reporting of life-threatening incidents.

Despite the advantages synthetic datasets offer, their generalization capabilities may vary across different tasks and real-world problem domains. Consequently, the subsequent chapters of this thesis shift their focus to solutions that mitigate this limitation by reducing labeled data for training. Specifically, the next chapters explore self-supervised representation learning, where a generic representation is initially learned from data and fine-tuned using limited-labeled data, as well as unsupervised learning, which eliminates the need for labeled data entirely. The goal is to develop techniques that generalize across diverse real-world scenarios.

References

[1] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," in *Proceedings of the ECCV*, 2018.

[2] J. Marín, D. Vázquez, D. Gerónimo, and A. M. López, "Learning appearance in virtual scenarios for pedestrian detection," in *Proceedings of CVPR*, 2010, pp. 137–144.

[3] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 746–753.

[4] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of CVPR*, 2020, pp. 2443–2451.

[5] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the CVPR*, June 2016.

[6] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?" in *Proceedings of ICCV*, 2021, pp. 10 849–10 859.

[7] C. Pramerdorfer, J. Strohmayer, and M. Kampel, "Sdt: A synthetic multi-modal dataset for person detection and pose classification," in *Proceedings of ICIP*, 2020, pp. 1611–1615.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of CVPR*, 2009, pp. 248–255.

[9] *General Data Protection Regulation (GDPR) – Official Legal Text*, https://gdpr-info.eu/, (Accessed on 03/02/2022).

[10] A. v. d. B. Paul Voigt, *The EU General Data Protection Regulation (GDPR) - A practical guide*. Cham: Springer International Publishing, 2017.

[11] M. Astrid, M. Zaheer, J.-Y. Lee, and S.-I. Lee, “Learning not to reconstruct anomalies,” in *Proceedings of the BMVC*, 2021.

[12] J.-H. Lee, M. Z. Zaheer, M. Astrid, and S.-I. Lee, “Smoothmix: A simple yet effective data augmentation to train robust classifiers,” in *Proceedings of the CVPRW*, June 2020.

[13] M. Astrid, M. Z. Zaheer, and S.-I. Lee, “Synthetic Temporal Anomaly Guided End-to-End Video Anomaly Detection,” in *Proceedings of ICCVW*, 2021, pp. 207–214.

[14] T. Karras, S. Laine, and T. Aila, “A style-based generator architecture for generative adversarial networks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4401–4410.

[15] V. V. Kniaz, V. A. Knyaz, J. Hladůvka, W. G. Kropatsch, and V. Mizginov, “Thermalgan: Multimodal color-to-thermal image translation for person re-identification in multispectral dataset,” in *Proceeding of ECCVW*, 2019, p. 606–624.

[16] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, “A comprehensive survey of image augmentation techniques for deep learning,” *Pattern Recognition*, vol. 137, p. 109347, 2023.

[17] H. M. Schlüter, J. Tan, B. Hou, and B. Kainz, “Natural synthetic anomalies for self-supervised anomaly detection and localization,” in *Proceedings of ECCV*, 2022.

[18] C. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-Supervised Learning for Anomaly Detection and Localization,” in *Proceedings of CVPR*, 2021, pp. 9664–9674.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” *Proceeding of CVPR*, 2017.

[20] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[21] B. Kawar, S. Zada, O. Lang, O. Tov, H. Chang, T. Dekel, I. Mosseri, and M. Irani, “Imagic: Text-based real image editing with diffusion models,” in *Proceedings of the CVPR*, 2023.

[22] H. Ravi, S. Kelkar, M. Harikumar, and A. Kale, “Preditor: Text guided image editing with diffusion prior,” *ArXiv*, 2023.

[23] R. Mokady, A. Hertz, K. Aberman, Y. Pritch, and D. Cohen-Or, “Null-text inversion for editing real images using guided diffusion models,” *ArXiv*, 2022.

[24] Y. Wang, J. Wang, G. Lu, H. Xu, Z. Li, W. Zhang, and Y. Fu, “Entity-level text-guided image manipulation,” *ArXiv*, 2023.

[25] R. Blythman, A. Elrasad, E. O’Connell, P. Kielty, M. O’Byrne, M. Moustafa, C. Ryan, and J. Lemley, “Synthetic thermal image generation for human-machine interaction in vehicles,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[26] F. Bongini, L. Berlincioni, M. Bertini, and A. Del Bimbo, “Partially fake it till you make it: Mixing real and fake thermal images for improved object detection,” in *Proceedings of the ACM International Conference on Multimedia*, 2021, p. 5482–5490.

[27] Unity Technologies, “Unity Perception package,” https://github.com/Unity-Technologies/com.unity.perception, 2020.

[28] B. O. Community, *Blender - a 3D modelling and rendering package*, 2018. [Online]. Available: http://www.blender.org

[29] K.-T. Lai, C.-C. Lin, C.-Y. Kang, M.-E. Liao, and M.-S. Chen, "Vivid: Virtual environment for visual deep learning," *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

[30] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. X. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, "Habitat 2.0: Training home assistants to rearrange their habitat," in *Proceedings of NeurIPS*, 2021, p. 251–266.

[31] Epic Games, "Unreal engine," 2019. [Online]. Available: https://www.unrealengine.com

[32] A. Acsintoae, A. Florescu, M.-I. Georgescu, T. Mare, P. Sumedrea, R. T. Ionescu, F. S. Khan, and M. Shah, "Ubnormal: New benchmark for supervised open-set video anomaly detection," in *Proceeding of CVPR*, 2022.

[33] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[34] *3DF ZEPHYR - photogrammetry software - 3D models from photos*, (Accessed on 03/01/2022). [Online]. Available: https://www.3dflow.net/3df-zephyr-photogrammetry-software

[35] J. K. Haas, "A history of the unity game engine," 2014.

[36] 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 *Proceedings of NeurIPS*, 2021.

[37] N. B. Joshi and S. Nalbalwar, "A fall detection and alert system for an elderly using computer vision and internet of things," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, 2017, pp. 1276–1281.

[38] N. Otanasap and P. Boonbrahm, "Pre-impact fall detection approach using dynamic threshold based and center of gravity in multiple kinect viewpoints," *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pp. 1–6, 2017.

[39] X. Li, T. Pang, W. Liu, and T. Wang, "Fall detection for elderly person care using convolutional neural networks," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–6.

[40] S. Kasturi and K.-H. Jo, "Human fall classification system for ceiling-mounted kinect depth images," in *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, 2017, pp. 1346–1349.

[41] Q. Feng, C. Gao, L. Wang, M. Zhang, L. Du, and S. Qin, "Fall detection based on motion history image and histogram of oriented gradient feature," in *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2017, pp. 341–346.

[42] H. Rajabi and M. Nahvi, "An intelligent video surveillance system for fall and anesthesia detection for elderly and patients," in *2015 2nd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, 2015, pp. 1–6.

[43] C.-J. Chong, W.-H. Tan, Y. C. Chang, M. Farid Noor Batcha, and E. Karuppiah, "Visual based fall detection with reduced complexity horprasert segmentation using superpixel," in *2015 IEEE 12th International Conference on Networking, Sensing and Control*, 2015, pp. 462–467.

[44] A. Yajai, A. Rodtook, K. Chinnasarn, and S. Rasmequan, "Fall detection using directional bounding box," in *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2015, pp. 52–57.

[45] R. Rucco, A. Sorriso, M. Liparoti, G. Ferraioli, P. Sorrentino, M. Ambrosanio, and F. Baselice, "Type and location of wearable sensors for monitoring falls during static and dynamic tasks in healthy elderly: A review," *Sensors*, vol. 18, no. 5, 2018.

[46] Y. Li, T. Banerjee, M. Popescu, and M. Skubic, "Improvement of acoustic fall detection using kinect depth sensing," in *2013 35th annual international conference of the IEEE Engineering in medicine and biology society (EMBC)*. IEEE, 2013, pp. 6736–6739.

[47] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[48] S. Bonderup, J. Olsson, M. Bonderup, and T. B. Moeslund, "Preventing drowning accidents using thermal cameras," in *Advances in Visual Computing*. Springer International Publishing, 2016, pp. 111–122.

[49] J. Liu, M. Philipsen, and T. Moeslund, "Supervised versus self-supervised assistant for surveillance of harbor fronts," in *VISAPP*, 2021, pp. 610–617.

[50] I. Nikolov, J. Liu, and T. Moeslund, "Imitating emergencies: Generating thermal surveillance fall data using low-cost human-like dolls," *Sensors*, vol. 22, no. 3, 2022.

[51] S. R. Nyman, C. Ballinger, J. E. Phillips, and R. Newton, "Characteristics of outdoor falls among older people: a qualitative study," *BMC Geriatrics*, vol. 13, no. 1, 2013.

[52] W. Li, T. H. M. Keegan, B. Sternfeld, S. Sidney, C. P. Quesenberry, and J. Kelsey, "Outdoor falls among middle-aged and older adults: a neglected public health problem." *American journal of public health*, vol. 96 7, pp. 1192–200, 2006.

[53] F. Kane, "Simulation of night-vision and infrared sensors," in *Game Engine Gems 2*, E. Lengyel, Ed. A K Peters, 2011, pp. 45–54.

[54] J. Xu, K. Läuger, R. Möller, K. Dransfeld, and I. H. Wilson, "Heat transfer between two metallic surfaces at small distances," *Journal of Applied Physics*, vol. 76, no. 11, pp. 7209–7216, 1994.

[55] N. Madan, M. S. N. Siemon, M. K. Gjerde, B. S. Petersson, A. Grotuzas, M. A. Esbensen, I. A. Nikolov, M. P. Philipsen, K. Nasrollahi, and T. B. Moeslund, "Thermalsynth: A novel approach for generating synthetic thermal human scenarios," in *Proceedings of the WACVW*, 2023, pp. 130–139.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of NIPS*, 2012, pp. 1106–1114.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

Chapter 4

Self-Supervised Representations Learning for Generic Visual Tasks

Self-supervised representation learning has gained popularity as a training approach for machine learning models. It offers the potential to learn useful representations [1, 2] that can be transferred to a wide range of downstream tasks. This has led to many exciting developments in the field, particularly in computer vision and natural language processing. In this chapter, we will explore the key concepts behind self-supervised learning and discuss a novel approach to train self-supervised models to improve the accuracy of downstream tasks.

1 Introduction

Self-supervised representation learning is a powerful approach that learns transferable representations [1, 2] through the use of unlabeled data. The main idea behind self-supervised representation learning is based on defining a *pretext task* on unlabeled data to learn representations [3, 4] that capture rich semantic features from the images. These representations can then be utilized for various *downstream tasks* such as image segmentation, and object classification.

In the field of natural language processing (NLP), self-supervised learning has gained significant popularity in recent years. The most common pretext objective in natural language processing (NLP) is to mask a word and predict the masked word based on the surrounding context [5], which is also the basis of large models like BERT [6]. This helps to capture the relationship across text without label data. The model learned with this objective generates a powerful representation that can be generalized for various linguistic tasks such as text translation [7, 8] and summarization [9].

In computer vision, the counterpart of masked language modeling (MLM) in NLP is masked image modeling (MIM). Models like BeiT [10] and PeCo [11] first leverage MIM to learn robust visual representations. BeiT is also considered as the BERT [6] equivalent in computer vision, while BERT itself introduced MLM in NLP. MIM [10–14] is an effective self-supervised learning (SSL) technique that involves masking out portions of an image and teaching a model to learn the missing context. MIM reconstructs masked patches of input images from visible ones. MIM generates highly semantic representations that can be effectively transferred to various downstream vision tasks. Recent research directions using MIM as a pretext task are inclined towards simplified algorithms such as MAE [12] and SimMIM [13] employing. MAE [12] shows that randomly masking 75% of the input tokens provides a strong pretext task in the image domain and leads to a strong representation. This thesis adopts MAE as the backbone due to its simplicity and generalizability.

The proposed approach in this chapter introduces a *novel masking module* that utilizes the knowledge of easy and hard tokens to adjust the complexity of the self-supervised pretext task. A representation is learned by introducing curriculum learning [15], where the backbone first learns to solve the easier task and then gradually raises the complexity. *Curriculum learning [15] is a well-established training strategy in machine learning, where data is organized to support learning processing in such a way that networks produce optimal results. Curriculum learning [15] includes sorting training samples based on complexity and incorporating them into the training schedule in increasing order of complexity.* The proposed approach in this chapter learns the generic representation by solving the masking task in increasing complexity. The results show that using curriculum masking as a pretext task results in a robust representation that outperforms the baseline by a significant margin. This chapter only contains preliminary results of our proposed approach. We plan to extend the experiments to different architectures and multiple visual tasks.

2 State-of-the-Art

Self-supervised Representation Learning. Some of the earlier works in the field of self-supervised learning (SSL) include layerwise pretraining [16, 17]. Bengio *et al.* [17] proposes a greedy layer-wise pretraining, where each layer of the neural network is trained one by one using an auto-encoder loss. An almost similar approach based on restricted Boltzman machines (RBMs) is proposed by Hinton *et al.* [16], which also incorporates layer-wise pretraining in the RBMs, and these RBMs are then stacked to generate a deep belief network. These methods, however, are replaced by simple strategies like denoising autoencoders [18] and deep canonically correlated autoencoders [19].

State-of-the-art (SOTA) literature in the field of self-supervised representational learning can broadly be categorized as deep metric-learning approaches, self-distillation architecture, canonical correlation analysis, and MIM. The objective of representational learning is to group similar samples together in the learned representation. The metric-learning-based approaches cluster similar images together by using objectives like contrastive loss [20] or triplet loss [21]. Early approaches in this category adopt the K-mean clustering method from classical machine learning to the feature space. Qian *et al.* [22] and Huang *et al.* [23] provide a deep clustering alternative and thus improve the representation.

Self-distillation architectures [24–26] usually contain two networks, both provided with a different transformation/view of the same sample and both networks are connected to a predictor which predicts one view from the other. These networks are limited by collapse issues, where the model learns to predict a constant value. BYOL [24] proposes a solution by applying a student-teacher [27] method, where one of the networks is updated with the moving average of the other network's weight.

SSL approaches based on canonical correlation analysis (CCA) [28] infer relationships by analyzing cross-covariance matrices between variables and samples. Some approaches from this category are SWAV [29] and VICReg [30]. SWAV [29] uses multiple views of the same data and clusters the representations of the views to encourage the model to learn semantic representations that capture different aspects of the data. VICReg [30] aims to optimize three objectives based on co-variance matrices of representations from two views: variance, invariance, and co-variance. The variance objective regularizes the variance along each dimension of the representation to prevent the model from collapsing. The invariance objective ensures that the two views are encoded similarly, meaning they have the same representation. The covariance objective encourages different dimensions of the representation to capture different features, making it more informative.

Image restoration [31–33] tasks play a significant role in SSL, where the learning goal is to restore missing or removed information in an image. Various methods [31–34] have been proposed to tackle this task, each with its own approach and pretext task. For example, the colorization-based method [35], first decolorizes an image and then trains the network to predict RGB values. Pathak *et al.* [33] adds the perturbation by masking out a large portion of an image and then the network learns to reconstruct the images by modeling the contextual information. Some approaches [32, 33] introduce "Jigsaw" as a pretext task by dividing an image into patches and shuffling their arrangement. The network will learn to restore the original arrangement and encode the contextual features. Doersch *et al.* [31] propose a pretext task by randomly sampling two patches from an image and learning to predict their relative positions. Out of all, the image restoration tasks [33, 34] shows superior representational transferability. This makes image inpainting and mask image modeling (MIM) the favourite tasks in SSL. However, with the development of the vision transformer, MIM with token masking [10, 12, 13] become more effective.

The idea of MIM (originally masking image patches) is now evolved into masked auto-encoding methods [6, 11–14, 36], in which the masked region is a union of mask tokens predicted using visual tokens in transformer-based backbones. This approach has proven to be highly effective in producing semantically meaningful representations for downstream tasks. As already discussed, MIM is directly related to MLM in BERT. Bao *et al.* [10] first proposed a language equivalent for BERT [6] for the image domain and found that applying this strategy directly to images is difficult as image patches can assume a large set of possible values, which is not suitable for a classification task. Therefore, Bao *et al.* [10] consider this problem in the image domain as a regression problem, by first using an auto-encoder to encode image patches as discrete tokens. This approach provides improvements over supervised learning on various downstream tasks. Dong *et al.* [11] further optimizes the codebook generation process by adding a perceptual loss. However, these approaches are rather complex as they involves a powerful auto-encoder network to create the discrete codebook. To mitigate the requirement of a codebook, Wei *et al.* [36] proposes a reconstruction target as HOG features [37] instead of discrete image tokens. However, these methods are replaced with simpler approaches like MAE [12] and SimMIM [13], which directly reconstruct the RGB values in the masked tokens rather than discrete token values generated by the auto-encoder in BeiT [10].

Inspired by the simplicity of MAE [12], some approaches [14, 38–40] modify the masking strategies. In MAE [12], random masking learns a generic and robust representation. ADIOS [38] proposes an adversarial masking strategy by training an image encoder with a masking network, resulting in an improved representation. Mixed MAE [39] increases the complexity of the pretext task by mixing two images using various approaches and enforces the network to undo the augmentation and learn the reconstruction simultaneously. SemMAE [40] introduces semantically-guided masking, where a separate part generator guides the MAE [12] to learn a robust representation. Kakogeorgiou *et al.* [14] proposes a student-teacher framework where the teacher network learns masking tokens to enhance the representation learned by the student model. Additionally, this chapter presents a method based on MAE [12] that generates multiple masks with different complexities and employs curriculum learning [15] to progressively tackle tasks with increasing mask complexity, improving the learned representation.

Curriculum Learning. Curriculum learning [15] is a machine learning technique where training data is arranged in a specific order to improve learning efficiency. The key elements of any curriculum strategy are the *curriculum criteria* and the *scheduling function*. The curriculum criteria determine the ordering of the data like easy-to-hard while the scheduling function determines when to update the training process.

The *curriculum criteria* determines the ordering of the data based on easy-to-hard [15, 41, 42] or hard-to-easy. In easy-to-hard scheduling, examples or tasks are presented in order of increasing difficulty, while in hard-to-easy scheduling, examples or tasks are presented in order of decreasing difficulty. The natural way of designing the curriculum is to arrange the training samples in increasing order of complexity. Several approaches [15, 41, 43–45] have adopted this strategy. This approach closely aligns with how humans learn, where they first tackle easier samples and gradually increase the complexity of the examples.

The curriculum can be generated manually or automatically. Some approaches [15, 46, 47] manually construct the curriculum based on measures like noise level [46], the degree of occlusion and shape complexity in images [15, 47], or other domain-specific properties. However, manually designing such a curriculum can be time-consuming and requires domain expertise.

To overcome this, automatic methods like self-paced learning (SPL) [42, 48], and teacher transfer method [49] are proposed. In self-paced learning [42], the difficulty of examples is measured based on example-wise training loss. The curriculum is then designed by gradually increasing the subset of data used for training. This is done starting with easy examples with low training loss and expanding to the entire dataset as the model improves.

Teacher transfer methods [49–51] involve a teacher model that estimates the difficulty of samples based on their performance across the samples. The teacher model transfers its knowledge about difficulty levels of examples to a student model. These methods can be classified based on the model capacity of the teacher model compared to the student model. For example, strong teacher [50] or same-level teacher [51].

In addition to the traditional curriculum design, some approaches [52, 53] propose hard-to-easy scheduling or anti-curriculum learning. For example, Shrivastava *et al.* [52] propose hard example mining (HEM) to improve object detection by emphasizing difficult examples. Braun *et al.* [53] employ anti-curriculum learning for ordering hard-to-easy examples based on the signal-to-noise ratio in an automatic speech recognition system.

The *scheduling function* plays a crucial role in determining when to update the training process based on the sorted samples in terms of their difficulty. Two main scheduling types can be distinguished: discrete and continuous. In discrete scheduling [15, 54], the data is initially sorted based on the curriculum criteria, arranging the samples from easy to hard. The training process begins with the easiest samples. After a certain number of epochs or convergence, the next set of samples is merged into the training subset. This process is repeated until all the sets have been processed, at which point the training process may stop or continue for a few additional epochs.

On the other hand, continuous scheduling [51, 55] uses a function to determine the proportion of easy training examples available at each epoch. This function maps the number of easy examples to a scalar value ranging from zero to one. One indicates that all easy training examples are available. The function should be non-decreasing, starting with a value greater than zero and ending at one. In literature, this function is also called pacing [51] or competence function [55].

In some domains, measuring examples' complexity can pose challenges, whether manual or automatic means. In such cases, curricula based on model complexity [56–58] or task complexity [59–62] can be particularly suitable. Curricula based on model complexity gradually increase the depth or capacity (size of hidden representations) of neural networks. On the other hand, curricula focused on task complexity increase the

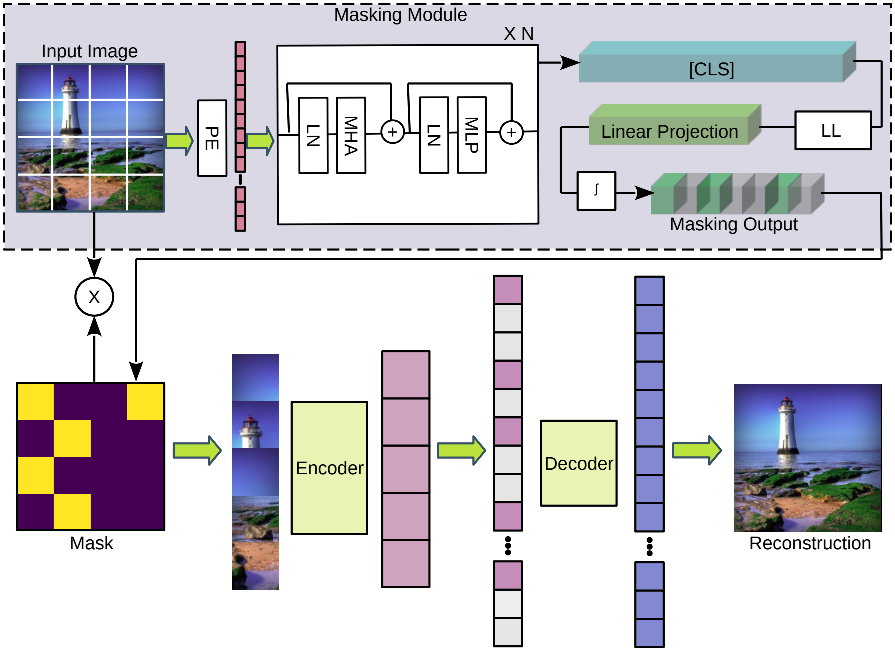


Fig. 4.1: The overall architecture to enable curriculum learning while using MAE as the backbone for self-supervised representation learning. The masking module generated the masking output based on the complexity of the task and provide the visible tokens to MAE based on this output.[Image and Caption are Taken from Paper F in this thesis]

complexity of the learning task by incorporating more complex objectives. This can involve initially training on simpler objectives and then gradually introducing more complex ones.

In this chapter, we present an approach to discrete curriculum scheduling. This is where the training process begins by processing all the easy samples before moving on to the more difficult examples.

3 Scientific Contribution

Chapters 1 and 2 of this thesis present two distinct approaches to learning specialized representations using labeled datasets. The first approach relies on human-annotated labels, while the second approach involves generating synthetic data with automatically generated annotations. Although manual annotations in the first approach are costly, the synthetic dataset in the latter approach does not consistently translate well across all visual tasks.

This chapter introduces a generic solution using self-supervised representation learning. It involves learning robust representations on large-scale datasets by defining a pretext task. This process, called model pretraining, enables the learned representation to be applicable to various visual tasks. However, for effective transfer of the learned representation to a specific visual task, the model needs to be fine-tuned with a minimal amount of labeled data. Paper F in this thesis proposes an approach based on curriculum learning to achieve a robust and transferable representation. This paper serves as a technical report within the thesis, presenting preliminary results that demonstrate improvements.

The network architecture of the proposed approach in paper F is shown in Figure 4.1. This architecture includes a learnable masking module and employs MAE [12] as the backbone for representational learning. Our approach adopts MIM as the pretext task, similar to MAE. However, unlike MAE, we employ a curriculum-based masking strategy instead of random masking [12]. The core concept of our framework revolves around curriculum learning, where the model initially addresses easier tasks and gradually advances towards more complex ones. To implement curriculum learning within our framework, the proposed masking module dynamically generates masks based on task complexity.

To design the curriculum, we propose a progressive loss function ($\mathcal{L}_{prog}$) (in Equation 4.1) as one of the objectives of our *novel masking module*. The network undergoes two stages of training based on $\mathcal{L}_{prog}$: positive or negative. In the first stage, when $\mathcal{L}_{prog}$ is positive, the masking module aligns its training direction with MAE to reduce the complexity of the pretext task. In the second stage, the masking module introduces adversarial training to increase the complexity of the pretext task. The progressive loss function is chosen to be identical to the pretext task's loss in MAE [12], which is calculated as the mean-squared error (MSE) between the normalized per-patch pixels of the reconstructed target ($\hat{\boldsymbol{I}}$) and the input image ($\boldsymbol{I}$), where η represents the number of epochs. The curriculum is updated after every η epoch to introduce increasingly complex tasks for representation learning.

$$\mathcal{L}_{\text{prog}}(\hat{\boldsymbol{I}},\boldsymbol{I})=\begin{cases}(\hat{\boldsymbol{I}}-\boldsymbol{I})^2, & \text{if } Epochs<\eta\\ -(\hat{\boldsymbol{I}}-\boldsymbol{I})^2, & \text{otherwise}\end{cases} \tag{4.1}$$

$$\mathcal{L}_{Total}=\lambda_{prog}\mathcal{L}_{prog}+\lambda_{Gauss}\mathcal{L}_{Gauss}+\lambda_{KL}\mathcal{L}_{KL}+\lambda_{div}\mathcal{L}_{div} \tag{4.2}$$

The proposed masking module has a total objective defined by Equation 4.2, where λ_{GL}, λ_{prog}, λ_{KL}, and λ_{div} are hyperparameters that determine the contributions of each loss term. The loss terms included in the objective are ($\mathcal{L}_{prog}$) for progressive learning, ($\mathcal{L}_{Gauss}$) for Gaussian loss, ($\mathcal{L}_{KL}$) for KL-divergence loss, and ($\mathcal{L}_{div}$) for diversity loss. Each loss term serves a specific purpose.

The Gaussian loss ensures that the masking module output is binary, with values of 0 for masked tokens and 1 for visible tokens. The KL-divergence loss makes sure that the ratio of masked pixels matches the pre-defined mask ratio. Diversity loss ensures each sample is assigned different masks, promoting diversity in training.

Method	Zero-shot	
	Acc@1	Acc@5
MAE (Baseline) [12]	39.2	61.5
Ours (MAE w/ Curriculum Masking)	**42.1**	**65.1**

Table 4.1: Illustrating the preliminary results obtained by including curriculum learning for representational learning using MIM as a pretext task. The preliminary results show a significant improvement and hence verify the idea.[Table is taken from paper F(technical report)]

Preliminary results, which verify that the idea of curriculum learning has the potential to improve the representation are shown in Table 4.1. The results obtained in the zero-shot setting, where we are not using any labeled data, surpass the baseline model, i.e., MAE [12] on both the metrics Acc@1 and Acc@5.

4 Summary

This chapter introduces an approach to curriculum learning to enhance generic representations learned by architectures like MAE on large-scale ImageNet datasets. Preliminary results of the image classification task are presented, demonstrating the potential transferability of the learned representations to other visual tasks with minimal labeled data. This chapter contributes as follows:

- Integration of curriculum learning into the MAE framework, where the curriculum is designed based on task complexity, gradually increasing from easy to hard.

- Introduction of a novel masking module that generates adaptive masks based on task complexity.

- Initial results indicate that incorporating curriculum learning during pre-training leads to more generic representations.

The proposed approach requires a lot of unlabeled data. This makes it suitable for scenarios like surveillance where extensive data exists but labelling it is impractical. It becomes feasible to learn the underlying semantics of the data by defining a suitable self-supervised task tailored to the surveillance domain.

The obtained representation has applicability across multiple visual tasks, but traditional pre-training is time-consuming and resource-intensive. In the following chapter, the thesis addresses this limitation by proposing self-supervised tasks for anomaly detection, eliminating the need for pre-training. However, the learned representation remains task-specific and lacks versatility for generic visual tasks.

References

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 119, 2020, pp. 1597–1607.

[2] I. Misra and L. v. d. Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the CVPR*, 2020.

[3] T. Hastie, R. Tibshirani, and J. Friedman, *Overview of Supervised Learning*. Springer, 2009, pp. 9–41.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, and Q. Liu, "TinyBERT: Distilling BERT for natural language understanding," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 4163–4174.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[7] T.-R. Chiang, Y.-P. Chen, Y.-T. Yeh, and G. Neubig, "Breaking down multilingual machine translation," in *Findings of the Annual Conference of the Association for Computational Linguistics (ACL)*, 2022.

[8] Q. Jiang, M. Wang, J. Cao, S. Cheng, S. Huang, and L. Li, "Learning kernel-smoothed machine translation with retrieved examples," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 7280–7290.

[9] H. Zhang, J. Cai, J. Xu, and J. Wang, "Pretraining-based natural language generation for text summarization," in *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2019, pp. 789–797.

[10] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *Proceeding of ICLR*, 2022.

[11] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo, "Peco: Perceptual codebook for bert pre-training of vision transformers," 2023.

[12] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 16 000–16 009.

[13] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the CVPR*, June 2022, pp. 9653–9663.

[14] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, A. Bursuc, K. Karantzalos, and N. Komodakis, "What to hide from your students: Attention-guided masked image modeling," in *Proceeding of the ECCV*, 2022, pp. 300–318.

[15] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009, p. 41–48.

[16] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, pp. 1527–1554, 2006.

[17] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy layer-wise training of deep networks,” in *Proceeding of the NeurIPS*, 2006.

[18] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” *Proceedings of the international conference on Machine learning (ICML)*, pp. 1096–1103, 2008.

[19] W. Wang, R. Arora, K. Livescu, and J. Bilmes, “On deep multi-view representation learning,” in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 37, 2015, pp. 1083–1092.

[20] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, “Signature verification using a "siamese" time delay neural network,” in *Proceeding of the Neural Information Processing Systems (NeurIPS)*, vol. 6, 1993.

[21] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, “Large scale online learning of image similarity through ranking,” in *Journal of machine learning research (JMLR)*, 2009.

[22] Q. Qian, Y. Xu, J. Hu, H. Li, and R. Jin, “Unsupervised visual representation learning by online constrained k-means,” *Proceeding of the CVPR*, pp. 16 619–16 628, 2022.

[23] P. Huang, P. Yao, Z. Hao, H. Peng, and L. Guo, “Improved constrained k-means algorithm for clustering with domain knowledge,” *Mathematics*, vol. 9, no. 19, p. 2390, 2021.

[24] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, B. Piot, k. kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent - a new approach to self-supervised learning,” in *Preceeding of the NeurIPS*, vol. 33, 2020, p. 21271–21284.

[25] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, “Emerging properties in self-supervised vision transformers,” in *Proceedings of the ICCV*, October 2021, pp. 9650–9660.

[26] X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of the CVPR*, 2021, pp. 15 750–15 758.

[27] L. Wang and K.-J. Yoon, “Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks,” *IEEE Transactions on Pattern Analysis Machine Intelligence (TPAMI)*, vol. 44, no. 06, pp. 3048–3068, 2022.

[28] H. Hotelling, *Relations Between Two Sets of Variates*, 1992, pp. 162–190.

[29] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9912–9924.

[30] A. Bardes, J. Ponce, and Y. LeCun, “VICReg: Variance-invariance-covariance regularization for self-supervised learning,” in *Proceeding of the International Conference on Learning Representations (ICLR)*, 2022.

[31] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proceedings of the ICCV*, December 2015.

[32] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, “Boosting self-supervised learning via knowledge transfer,” in *Proceedings of the CVPR*, June 2018.

[33] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the CVPR*, June 2016.

[34] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, "A cookbook of self-supervised learning," 2023.

[35] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proceedings of the ECCV*, 2016.

[36] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceedings of the CVPR*, 2022, pp. 14 668–14 678.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceeding of the CVPR*, vol. 1, 2005, pp. 886–893.

[38] Y. Shi, N. Siddharth, P. H. Torr, and A. R. Kosiorek, "Adversarial masking for self-supervised learning," in *Proceeding of the International Conference on Machine Learning (ICML)*, 2022.

[39] K. Chen, Z. Liu, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, "Mixed autoencoder for self-supervised visual representation learning," 2023.

[40] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng, "Semmae: Semantic-guided masking for learning masked autoencoders," in *Proceedings of the NeurIPS*, 2022.

[41] R. T. Ionescu, B. Alexe, M. Leordeanu, M. C. Popescu, D. P. Papadopoulos, and V. Ferrari, "How hard can it be? estimating the difficulty of visual search in an image," *Proceeding of the CVPR*, pp. 2157–2166, 2016.

[42] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proceeding of the Neural Information Processing Systems (NeurIPS)*, vol. 23, 2010.

[43] X. Chen and A. K. Gupta, "Webly supervised learning of convolutional networks," *Proceeding of the ICCV*, pp. 1431–1439, 2015.

[44] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum learning of multiple tasks," *Proceeding of the CVPR*, pp. 5492–5500, 2014.

[45] Y. Shi, M. Larson, and C. M. Jonker, "Recurrent neural network language model adaptation with curriculum learning," *Comput. Speech Lang.*, vol. 33, pp. 136–154, 2015.

[46] S. Braun, S.-C. Liu, and D. Neil, "A curriculum learning method for improved noise robustness in automatic speech recognition," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2018.

[47] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas, "Curriculum deepsdf," in *Proceedings of the ECCV*, 2020, pp. 51–67.

[48] Q. Zhao, D. Meng, L. Jiang, Q. Xie, Z. Xu, and A. Hauptmann, "Self-paced learning for matrix factorization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[49] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceeding of the International Conference on Machine Learning(ICML)*, 2017, p. 2304–2313.

[50] D. Weinshall, G. Cohen, and D. Amir, “Curriculum learning by transfer learning: Theory and experiments with deep networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2018, pp. 5238–5246.

[51] G. Hacohen and D. Weinshall, “On the power of curriculum learning in training deep networks,” in *Proceedings of International Conference on Machine Learning (ICML)*, 2019, pp. 2535–2544.

[52] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in *Proceedings of the CVPR*, 2016, pp. 761–769.

[53] S. Braun, D. Neil, and S.-C. Liu, “A curriculum learning method for improved noise robustness in automatic speech recognition,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 548–552.

[54] V. I. Spitkovsky, H. Alshawi, and D. Jurafsky, “From baby steps to leapfrog: How “less is more” in unsupervised dependency parsing,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010, pp. 751–759.

[55] E. A. Platanios, O. Stretcu, G. Neubig, B. Póczos, and T. M. Mitchell, “Competence-based curriculum learning for neural machine translation,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[56] P. Morerio, J. Cavazza, R. Volpi, R. Vidal, and V. Murino, “Curriculum dropout,” *Proceeding of the ICCV*, pp. 3564–3572, 2017.

[57] T. Karras, T. Aila, S. Laine, and J. Lehtinen, “Progressive growing of GANs for improved quality, stability, and variation,” in *Proceeding of the International Conference on Learning Representations (ICLR)*, 2018.

[58] S. Sinha, A. Garg, and H. Larochelle, “Curriculum by smoothing,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, p. 21653–21664.

[59] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, “Reverse curriculum generation for reinforcement learning,” in *Proceedings of the Conference on Robot Learning (CoRL)*, vol. 78, 2017, pp. 482–495.

[60] W. Lotter, G. Sorensen, and D. Cox, “A multi-scale cnn and curriculum learning strategy for mammogram classification,” in *Proceeding of the Deep Learning in Medical Image Analysis (DLMIA) and Multimodal Learning for Clinical Decision Support (ML-CDS)*, 2017, pp. 169–177.

[61] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. Kakadiaris, “Curriculum learning for multi-task classification of visual attributes,” in *Proceedings of the ICCVW*, 10 2017, pp. 2608–2615.

[62] Y. Zhang, P. David, and B. Gong, “Curriculum domain adaptation for semantic segmentation of urban scenes,” *Proceedings of the ICCV*, pp. 2039–2049, 2017.

Chapter 5

Unsupervised Learning for Anomaly Detection

Unsupervised learning has emerged as a prominent area of research in computer vision. It offers promising possibilities for tackling diverse tasks and challenges by allowing models to learn patterns and structures directly from input data without the need for explicit supervision. In this chapter, we explore the application of unsupervised learning specifically for anomaly detection.

1 Introduction

Anomaly detection is stated as the identification of rare or abnormal instances within a dataset. By leveraging the power of deep neural networks trained unsupervised, the model can learn the intrinsic patterns of normal data and effectively detect deviations or anomalies. This approach eliminates manual annotation of anomalies. This makes it relevant and applicable across various domains, including intrusion detection, fraud detection, defect detection, and medical diagnostics.

The anomaly detection task in computer vision is challenging due to the subjective nature of defining anomalies, which can vary depending on the context. Normal behavior in one scenario may be considered abnormal in another. For example, workers wearing safety helmets at a construction site are normal, but abnormal in other contexts. This task requires a high level semantic understanding of images and videos to accurately capture and interpret behavior patterns.

In the domain of image anomaly detection, the literature primarily focuses on solving the task in unsupervised settings [1–8]. On the other hand, video anomaly detection is addressed in both weakly supervised [9–13] and unsupervised settings [4, 5, 14–24]. This PhD thesis focuses on solving anomaly detection using unsupervised learning approaches for both image and video data. Unsupervised anomaly detection tasks adopt the framework of one-class classification. In one-class classification, the training process centres around a singular class called the "normal" class. The primary goal is to cultivate a model's capability to effectively discern instances belonging to the normal class. In addition, it is to identify and categorise observations that deviate beyond the predefined threshold of normality as anomalies.

To achieve normal class classification, models typically utilize a self-supervised task, and the loss incurred from this task serves as a classification metric. Commonly used self-supervised tasks for anomaly detection include reconstruction [4, 5, 7, 14–20, 25–28] and mask prediction [4, 16, 22, 29–32]. The mask information in these tasks can take various forms, such as predicting future frames [16] and completing missing video frames [32] for video data, and restoring attributes [4] and inpainting missing regions [33]. This chapter presents approaches that incorporate masked pixels prediction at the core architectural level. Two different neural blocks that focus on reconstructing masked pixels based on the surrounding context are discussed later in this chapter. The proposed blocks can be integrated with different neural network architectures at different levels to improve context modeling for anomaly detection task. These blocks are designed in both 2D and 3D (as shown in Figure 5.1) to encode both spatial and spatio-temporal contexts respectively.

In addition, this chapter introduces an approach that utilizes trajectories as long-range temporal cues for anomaly detection. Trajectories capture objects' motion characteristics, providing valuable information for behavior analysis and object interactions. These trajectories are obtained by running a tracker on the video sequence in the anomaly detection datasets. By analyzing motion patterns and interactions among multiple object trajectories, it becomes possible to distinguish between normal and abnormal trajectories based on their characteristics.

While this chapter focuses on using centre coordinates of the trajectories for simplicity, the approach can be extended to include additional characteristics such as feature embeddings and bounding box information of the objects. By leveraging centre coordinates, the system achieves faster response times in detecting anomalies and improves efficiency. However, relying solely on trajectory information may be insufficient for complex video data. To overcome this limitation, we enhance the approach by incorporating spatial cues based on feature embeddings extracted from individual frames. This integration of spatial cues complements trajectory-based anomaly detection, enhancing the system's overall performance in analyzing complex video data.

2 State-of-the-Art

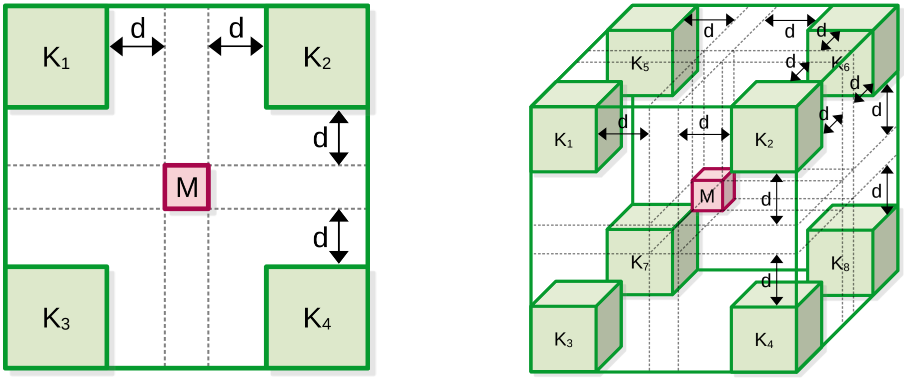
Anomaly Detection. State-of-the-art (SOTA) approaches for unsupervised anomaly detection can be classified into various categories, i.e., Dictionary learning methods [2, 34–38], Distance-based approaches [3, 21, 39–50], Probabilistic models, Change detection frameworks [51–54], and Reconstruction-based methods [4, 5, 7, 14–20, 25–28]. Each category offers specific techniques to address anomaly detection challenges.

Dictionary learning methods [2, 34–38] construct a dictionary or set of basis vectors using normal instances. Instances that cannot adequately be represented by the learned dictionary are classified as abnormal. Distance-based approaches [3, 21, 39–50] learn a distance function that distinguishes between normal and abnormal samples. Change detection frameworks [51–54] directly analyze test videos by quantifying the level of variation between the current frame and the preceding frames to detect abnormal frames. Probabilistic models [6, 55–63] learn probability density function of normal data and identify instances that deviate from the learned distribution as abnormal. Reconstruction-based methods [4, 5, 7, 14–20, 25–28] emphasize on learning the reconstruction of normal examples. Anomalies are identified by assessing the magnitude of the reconstruction error, as they often exhibit larger errors than normal instances. These categories encompass diverse strategies used in anomaly detection, each offering distinct advantages and drawbacks. A comprehensive understanding of these approaches plays a key role in advancing the development of robust anomaly detection methods within computer vision. Inspired by its dominance, this chapter presents approaches based on reconstruction methods.

Reconstruction-based methods have gained prominence in anomaly detection, particularly in image and video scenarios. These approaches, including auto-encoders and generative adversarial networks (GANs) to learn a latent manifold that accurately represents the distribution of normal data [19, 22, 27, 64]. In unsupervised video anomaly detection, some methods incorporate motion cues by reconstructing optical flow [16, 64, 65] or gradients [21], enabling motion-based anomalies. However, reconstruction-based frameworks show a tendency to generalize, by reconstructing even abnormal frames with low errors.

To address the generalization issue, researchers introduce memory modules into auto-encoders [14, 19, 64] and incorporate pseudo anomalies during training [14, 19, 64, 66]. The inclusion of memory modules in auto-encoders increases computational complexity as it requires an additional module to store prototypes of normal samples during training. Additionally, these approaches are constrained by memory size. Pseudo anomalies are generated through augmentation [66, 67] or considering out-of-distribution samples (e.g., flowers, anime, texture) as fake anomalies [65]. The methods usually employ data augmentation techniques to generate pseudo anomalies such as skipping frames in a video [67], combining poor reconstructions of a frame from earlier checkpoints [66], synthesizing natural anomalies using Poisson editing [68]. The approaches based on pseudo anomalies train networks using gradient descent on normal data and gradient ascent on pseudo-abnormal data. However, this adversarial training can increase convergence time and may lead to instability issues when balancing between gradient ascent and gradient descent.

In this chapter, we propose three approaches to unsupervised anomaly detection in images and videos. The first approach involves modelling only spatial contexts using our proposed masked convolutional block in 2D. The second approach includes masking both spatial and spatio-temporal contexts by extending the block to 3D as well. The third approach presents modelling of spatial and temporal contexts individ-



(a)

(b)

Fig. 5.1: (a): 2D masked convolution proposed in Paper C to capture spatial context, (b): 3D masked convolution proposed in paper D to capture spatio-temporal context [Figures are tasked from [24, 69]]

ually and finally combines scores for video anomaly detection. The next parts of this section provide more details about the individual concept.

Learning Spatial Context. Learning spatial context in deep neural networks refers to the ability of the network to capture and understand relationships between spatially adjacent features in an image. Convolutional neural networks (CNNs) have long been dominant networks for context modeling in computer vision, but the landscape shifted when Vaswani *et al.* [70] introduced the self-attention mechanism. This breakthrough sparked a wave of research exploring neural architectures that heavily rely on attention as the primary mechanism for modeling context.

CNNs are designed to exploit local connectivity and shared weights across the input space, allowing them to effectively learn spatial hierarchies of features. However, to fully leverage the spatial context, additional techniques such as using larger receptive fields [71], dilated/atrous convolution [71] (induce gap between the convolutional kernel) and spatial pyramid pooling [72], are proposed. Each of these approaches has its own pros and cons and therefore is applicable to a specific situation. CNNs use features from individual or combined multiple scales to adaptively capture local to global context in an image. For example, spatial pyramid pooling [72] helps CNNs capture multi-scale spatial context. It involves dividing input feature maps into sub-regions of different sizes and pooling features separately within each sub-region. This allows the network to capture contextual information at multiple scales, which allows it to handle objects of various sizes within an image. Furthermore, attention mechanisms such as BAM [73] and CBAM [74] can be employed to explicitly model the importance of various spatial locations. The attention mechanism enables the network to dynamically weigh the contribution of different spatial locations based on their relevance to the task at hand. These approaches enable the network to effectively capture and utilize the spatial relationships between features, leading to improved performance in various computer vision tasks such as object recognition [75, 76], semantic segmentation [77, 78], and anomaly detection [21, 23, 24, 79, 80].

Later on, these methods are replaced by architectures such as vision transformers [81–92]. These models are based on the self-attention idea proposed by Vaswani *et al.* [70]. These models are used widely now in the field of computer vision, mostly because of their remarkable performance on a wide range of tasks including object recognition [83, 87, 88], object detection [81, 91, 92], image generation [86, 89, 90], and anomaly detection [30, 31, 93, 94]. Transformer models in these cases capture long-range dependencies. The self-attention mechanisms enable them to attend different parts of the input and capture contextual relationships between them.

Learning Spatio-Temporal Context. Modeling spatial context is essential for image-based tasks, as it enables effective understanding and analysis of visual information. However, when dealing with video data, a more comprehensive approach is required to capture the spatio-temporal context accurately. Therefore, it is crucial to employ suitable modeling techniques that account for both spatial and spatio-temporal contexts, depending on the characteristics of anomalies present in the dataset. Some approaches [16, 21, 23, 64, 65, 67] in the domain of unsupervised video anomaly detection are modeling both spatial and spatio-temporal contexts to capture both motion-related and spatial anomalies. However, a large set of approaches in this domain include optical flow [16, 64, 65] as temporal cues and image features as spatial cues. Some other examples of approaches to model spatio-temporal aspects apart from optical flow are Ionescu *et al.* [21] included gradients, Yu *et al.* [32] learning to complete videos and includes spatio-temporal context for video anomaly detection. Wan *et al.* [95] designed a spatio-temporal jigsaw puzzle solving based on four consecutive frames. Some approaches [22, 96] also use 3D convolution, which is a computationally expensive operation. However, these methods address this challenge by selectively processing foreground objects (based on detection bounding boxes) within video frames, while excluding the background.

Other techniques to encode spatio-temporal cues include encoding the temporal context by using RNNs [97] and its variants GRU [98] and LSTM [97], which is often combined with CNNs to model spatio-temporal context. Some of the more sophisticated approaches include I3D [99] and C3D [100], which better model intricate spatio-temporal interactions. It is worth noting that the inclusion of a more intense spatio-temporal context is also computationally expensive and therefore sometimes avoided to save computation time. However, I3D and C3D features are often used in weakly supervised anomaly detection domains. This is where video datasets like UCF Crime [9], XDViolance [101] are more complex. Anomalies in datasets like explosions and shoplifting are not only motion-based but also involve intricate spatio-temporal interactions. Detecting such anomalies necessitates intense spatio-temporal context modeling to capture dynamic relationships over time. On the other hand, existing datasets in the domain of unsupervised anomaly detection, e.g., Avenue [37] and Shanghai-Tech [16], are biased towards motion-based anomalies such as running, jumping, or similar behaviors, which could simply be solved by including optical flow. Therefore, these datasets are limited in terms of being representative, since they might include more complex spatio-temporal interactions in abnormal instances.

Masking for Anomaly Detection. Some existing studies [4, 16, 22, 29–32] incorporates masked input prediction as an auxiliary task for anomaly detection, forming a distinct subset of reconstruction-based methods. For instance, Liu *et al.* [16] presents a GAN [102] framework that predicts future frames based on a limited number of past frames, classifying anomalies based on the prediction error. Another GAN-based approach [33] simultaneously detects and localizes anomalies through inpainting. In this method, the generator network is trained to fill in masked patches in the input image (also known as inpainting). Simultaneously, the discriminator network is trained to distinguish between normal and abnormal patches inpainted by the generator. Interestingly, the inpainting task has also been explored in combination with vision transformers as shown in [31]. Haselmann *et al.* [29] proposes to erase a rectangular subsection from the center of the image and classify anomalies based on the interpolation error. Fei *et al.* [4] proposes an Attribute Restoration Network (ARNet), where they mask attributes (like color and orientation) of an image and the model learns to restore those attributes. Building upon the success of masked auto-encoders [103], Jiang *et al.* [30] introduces a masked Swin Transformer [104] designed specifically for inpainting masked regions. To overcome transformer data requirements, this method simulates artificial anomalies. These approaches propose masking tasks for images to improve context modeling.

Video masking can be integrated at a spatiotemporal level, where, rather than masking pixels from a single frame, entire frames are masked. Based on this approach, Georgescu *et al.* [22] first generates spatio-temporal cubes using a detection bbox for each object in a video and then proposes middle box masking as one of the auxiliary tasks in their multi-task learning framework. Liu *et al.* [16], Yu *et al.* [32] employs the Cloze task [105], where specific frames from videos are masked and the model learns to complete videos using available frames.

Inspired by the success of masking tasks, this chapter presents approaches for unsupervised anomaly detection, by introducing the concept of masking at the core architectural level in form of blocks (referred to as SSPCAB and SSMCTB in this chapter). These blocks improve discrimination between normal and abnormal pixels by predicting masked pixels.

Trajectories for Anomaly Detection. In the domain of surveillance, anomaly detection based on trajectories is relatively limited. Some notable instances include the detection of anomalous trajectories in the context of traffic surveillance [106, 107]. These approaches are mostly based on clustering of trajectories using hand-crafted features and distance measures between trajectories. The cluster with small support, in this case, is considered anomalous.

Unsupervised anomaly detection on human trajectories is still an under-explored area, mostly because of the biasing towards optical flow as discussed earlier in this section. The trajectories however are considered as a special case of time-series, and there exist some approaches for anomaly detection on time-series data [108, 109]. The earlier approaches in this area are based on statistical methods: k-mean clustering [108], distance-measures [110]. These approaches are later replaced by deep neural networks [111–113] using auto-encoders based on RNNs [114] and LSTM [113] to encode these time-series. However, anomaly detection on human trajectories is a difficult task as there are multiple external factors that impact human motion such as context, density, social interaction, and many others.

Bouritsas *et al.* [115] constructs trajectories by selecting key points from human-skeleton trajectories and detect anomalies based on these trajectories. However, this framework requires precise key-point detection, which is considered a difficult task. To simplify the task, this chapter presents a method of constructing those trajectories based on a single point, i.e., the center-of-mass. The trajectory constructed using this approach are computationally efficient. The existing literature considers some approaches for trajectory prediction, which can be employed as a surrogate task for unsupervised anomaly detection. Approaches like Social LSTM [116] and Social GAN [117] focus on predicting trajectories by including multiple cues such as social interaction, past motion, and environment dynamics. Social LSTM [116] introduces a social pooling layer, which considers the interaction among different trajectories. The Social GAN [117], later extends the concept with GAN-based architecture and introduces variety loss to predict a diverse set of trajectories. This chapter proposes an anomaly detection framework using trajectories, where anomalies are classified based on prediction error. Our approach also considers social interaction and variety loss, similar to the Social GAN [117] to detect anomalous trajectories.

Overall, two classes of methods for anomaly detection are discussed in this chapter: a) First category illustrates improvement by modeling the spatial and spatio-temporal context using *novel neural blocks*. The proposed block encapsulates the capability to reconstruct the missing information based on the surrounding context and can be integrated easily with other SOTA methods. b) Second category utilizes social interaction among trajectories as a crucial temporal cue and integrates it with approaches using mostly using spatial cues for anomaly detection.

3 Scientific Contributions

Preceding chapters of this thesis focus on two distinct learning methodologies. Initial approaches entailed supervised learning, where the development of robust models for specific tasks heavily relied on annotated datasets. The second approach aimed at acquiring representative features, typically necessitating fine-tuning using a limited

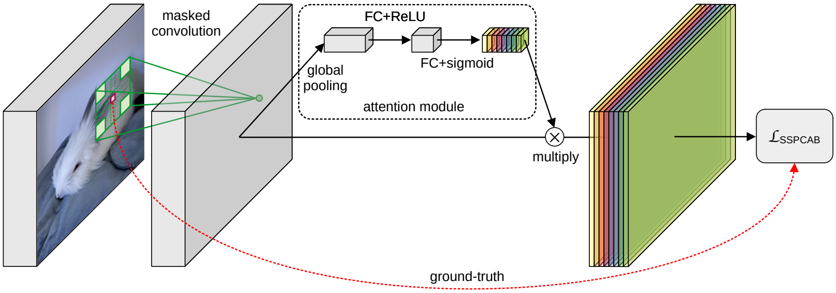


Fig. 5.2: Our self-supervised predictive convolutional attentive block (SSPCAB). For each location where the dilated convolutional filter is applied, the block learns to reconstruct the masked area using contextual information. A channel attention module performs feature recalibration by using global information to selectively emphasize or suppress reconstruction maps. Best viewed in color.[Image and caption are taken from [24]]

Method	AUROC		Localization AP
	Detection	Localization	
DRAEM [27]	41.06	42.40	45.41
DRAEM + SSPCAB [24]	44.19	46.66	46.89
DRAEM + SSMCTB (Ours)	**50.27**	**53.98**	**50.75**
NSA [68]	53.66	74.90	61.09
NSA + SSPCAB [24]	54.91	75.30	62.37
NSA + SSMCTB (Ours)	**60.09**	**77.09**	**64.55**
3D DRAEM [27]	43.74	44.12	45.97
3D DRAEM + 3D SSMCTB (Ours)	**53.70**	**58.47**	**52.79**

Table 5.1: Detection AUROC and localization AUROC/AP (in %) of two state-of-the-art methods [27, 68] on BRATS, before and after alternatively adding SSPCAB and SSMCTB. Additional results obtained by converting DRAEM to use 3D convolutions and integrating the 3D SSMCTB are also reported. The best result for each model and each performance measure is highlighted in bold. [Table and caption is taken from [69]]

labeled dataset to address visual tasks. In contrast, the current chapter tackles the challenge of unsupervised anomaly detection, eliminating the need for labeled data. Instead, we adopt a one-class classification framework, training the model only on normal samples and evaluating it on both normal and abnormal samples during testing. This unsupervised approach allows for additional flexibility and applicability in real-world scenarios where annotations are impractical or infeasible. The research presented in Papers C, D, and E of this thesis addresses the unsupervised anomaly detection task.

Papers C and D are closely interconnected as they both introduce masked convolution as a neural block and explore its potential for enhancing contextual modeling. This neural block consists of three key components: masked convolution, an attention module, and self-supervised loss. In this chapter, we propose methodologies that

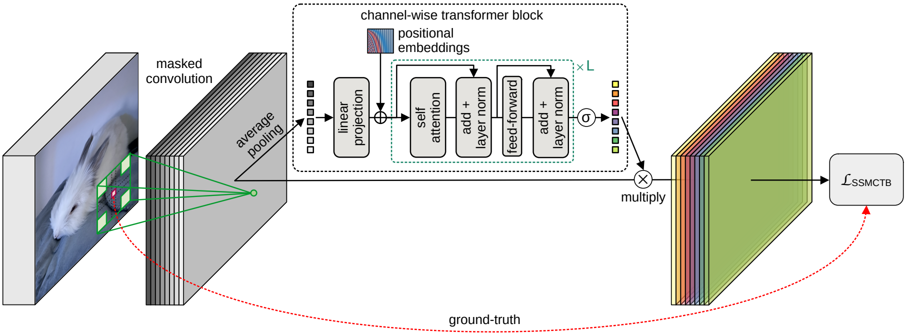


Fig. 5.3: An overview of our self-supervised masked convolutional transformer block (SSMCTB). At every location where the masked filters are applied, the proposed block has to rely on the visible regions (sub-kernels) to reconstruct the masked region (center area). A transformer module performs channel-wise self-attention to selectively promote or suppress reconstruction maps via a set of weights returned by a sigmoid (σ) layer. The block is self-supervised via the Huber loss ($\mathcal{L}_{\text{SSMCTB}}$) [118] between masked and returned activation maps. Best viewed in color.[Image and caption are taken from [69]]

Method	AUC	
	Micro	Macro
Park *et al.* [19]	53.2	66.5
Park *et al.* [19] + SSPCAB	53.6	**66.6**
Park *et al.* [19] + SSMCTB (Ours)	**58.9**	**66.6**

Table 5.2: Micro and macro AUC scores (in %) on Thermal Rare Event, obtained while alternatively including SSPCAB [24] and SSMCTB into the method of Park *et al.* [19]. [Table and caption are taken from [69]]

leverage spatial and spatio-temporal contexts for anomaly detection tasks. Initially, Paper C uses 2D masked convolution (Figure 5.1 (a)) to capture spatial context exclusively. Later, in Paper D, we extend this concept to 3D masked convolution (Figure 5.1 (b)) to incorporate spatio-temporal context, enabling the model to consider dependencies on adjacent frames in videos. Additionally, Paper E suggests incorporating object trajectories as an additional cue within anomaly detection frameworks to capture long-range dependencies.

In Paper C, we utilize aforementioned components, employing a 2D masked convolution followed by a channel attention module using Squeeze-and-Excitation (SE) module [119]. The self-supervised loss is measured using mean square error (MSE) (Figure 5.2). We refer to this block as the self-supervised predictive convolution attentive block (SSPCAB). By integrating SSPCAB with six different baselines, consisting of four for video anomaly detection and two for image anomaly detection, we achieve promising results. Table 5.4 presents the performance of SSPCAB in video anomaly detection on widely used public datasets such as Avenue [37] and Shanghaitech [16]. The evaluation metrics include area under the curve (AUC), RBDC [41], and TBDC [42]. The results consistently outperform baselines, establishing SSPCAB as the new state-of-the-art (SOTA) for certain metrics on these datasets.

Moreover, we demonstrate the integration of SSPCAB with two baselines for image anomaly detection on MvTec dataset [1] in Table 5.3. The results measures anomaly detection and location accuracy using the area under the ROC curve (AUROC) and average precision (AP) metrics. The results consistently outperform the baseline, further emphasizing SSPCAB's effectiveness in image anomaly detection scenarios.

In Paper D, we incorporate either 2D or 3D masked convolution, followed by a self-attention mechanism based on a transformer block. Huber loss replaces the self-supervised loss in this case, offering increased robustness against outliers (Figure 5.3). We refer to this block as the self-supervised masked convolutional transformer block (SSMCTB). SSMCTB is integrated with eight baselines for image and video anomaly detection, consisting of six for video anomaly detection and two for image anomaly detection. We further expand the application domain to include thermal, and medical data. The integration of SSMCTB with six different baselines for video anomaly detection in the RGB domain is presented in Table 5.4. Similar to Paper C, the results in Table 5.4 showcase the performance on Avenue [37] and ShanghaiTech [16] datasets. Additionally, results of integrating SSMCTB with a single baseline for video anomaly detection in the thermal domain are shown in Table 5.2. In Paper D, we proposed a novel dataset for detecting anomalies in the thermal domain. This dataset is generated by annotating one week video from the season-in-drift dataset [120].

Furthermore, we demonstrate the integration of SSMCTB with two baselines from image anomaly detection on the MvTec dataset in the RGB domain in Table 5.3. Finally, the results of integrating SSMCTB with two baselines from image anomaly detection on a medical dataset called BraTs [121] are presented in Table 5.1. While the BraTs dataset is initially proposed for brain tumor segmentation, we consider tumors as anomalies in this case. The evaluation metrics used in this paper are same as those in Paper C for both video and image anomaly detection datasets. The results consistently show that SSMCTB provides further improvement over SSMCTB in most cases, showcasing its effectiveness in various domains.

Paper E incorporates long-range dependencies as temporal cues into learning algorithms for unsupervised video anomaly detection. This paper introduces these long-range temporal relations through trajectories. These trajectories are constructed by applying a pre-trained tracker to the publicly available Avenue [37] and ShanghaiTech [16] datasets, which are widely used in the field of unsupervised video anomaly detection. Each track is generated by combining the center points of the detected bounding boxes across the temporal dimension.

To detect anomalies in trajectories, paper E generalizes a well-known trajectory prediction approach called SocialGAN [117]. Inspired by Liu et al's future-frame prediction method [16], the underlying idea is that the prediction error of abnormal trajectories, which were not encountered during training, is greater than that of normal trajectories. The prediction of each trajectory takes into account a few past samples

	Class	Detection						Localization								
		DRAEM [27]			NSA (logistic) [68]			DRAEM [27]						NSA (logistic) [68]		
		AUROC			AUROC			AUROC			AP			AUROC		
		Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB
Texture	Carpet	97.0	**98.2**	96.8	95.6	**97.5**	96.1	95.5	95.0	**95.8**	53.5	**59.4**	55.2	95.5	**97.5**	95.6
	Grid	99.9	**100**	**100**	99.9	99.9	**100**	**99.7**	99.5	**99.7**	65.7	61.1	**69.7**	**99.2**	**99.2**	**99.2**
	Leather	**100**	**100**	**100**	99.9	99.9	**100**	98.6	**99.5**	97.6	75.3	**76.0**	65.5	99.5	99.5	**99.6**
	Tile	99.6	**100**	**100**	**100**	**100**	**100**	99.2	**99.3**	**99.3**	92.3	95.0	**95.7**	**99.3**	99.2	99.1
	Wood	99.1	99.5	**100**	97.5	97.7	**97.8**	96.4	**96.8**	94.8	**77.7**	77.1	75.6	90.7	90.4	**93.5**
Object	Bottle	99.2	98.4	**99.4**	**97.7**	**97.7**	**97.7**	99.1	98.8	**99.2**	86.5	87.9	**89.9**	98.3	98.3	**98.4**
	Cable	91.8	**96.9**	94.1	94.5	95.6	**96.1**	94.7	**96.0**	95.5	52.4	57.2	**61.6**	96.0	96.6	**97.5**
	Capsule	98.5	**99.3**	97.1	95.2	95.4	**95.5**	**94.3**	93.1	93.4	49.4	50.2	**52.0**	97.6	97.2	**97.9**
	Hazelnut	**100**	**100**	**100**	94.7	94.2	**97.1**	99.7	**99.8**	99.5	**92.9**	92.6	89.1	97.6	**97.9**	**97.9**
	Metal Nut	98.7	**100**	**100**	98.7	99.0	**99.5**	**99.5**	98.9	99.3	96.3	**98.1**	94.7	98.4	**98.6**	98.3
	Pill	98.9	**99.8**	**98.8**	99.2	99.2	**99.5**	**97.6**	97.5	97.4	48.5	**52.4**	46.9	98.5	**98.8**	98.4
	Screw	93.9	97.9	**99.0**	90.2	**91.1**	90.4	97.6	**99.8**	99.5	58.2	**72.0**	70.1	**96.5**	96.2	96.4
	Toothbrush	**100**	**100**	**100**	**100**	**100**	**100**	98.1	98.1	**99.0**	44.7	51.0	**69.0**	94.9	95.3	**95.4**
	Transistor	93.1	92.9	**96.0**	95.1	95.6	**96.2**	**90.9**	87.0	89.1	**50.7**	48.0	45.8	88.0	87.1	**88.3**
	Zipper	**100**	**100**	**100**	99.8	99.8	**99.9**	98.8	**99.0**	**99.0**	**81.5**	77.1	76.5	94.2	94.5	**94.7**
	Overall	98.0	**98.9**	98.7	97.2	97.5	**97.7**	**97.3**	97.2	97.2	68.4	70.3	**70.5**	96.3	96.4	**96.7**

Table 5.3: Detection AUROC and localization AUROC/AP (in %) of two state-of-the-art methods [27, 68] on MVTec AD, before and after alternatively adding SSPCAB and SSMCTB. The best result for each model and each performance measure is highlighted in bold. [Table and caption are taken from [69]]

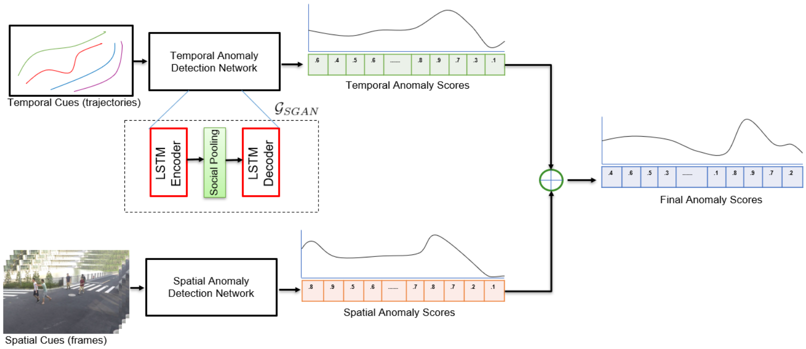


Fig. 5.4: The network architecture of the proposed method in paper E combines long-range temporal cues from trajectories and spatial cues from SOTA methods. Anomaly scores are generated by weighting the outputs of the spatial and temporal branches. [Figure is taken from [23]]

and social interaction among sub-trajectories to predict future samples. Abnormalities are inferred to occur in situations involving unusual social interactions, resulting in higher prediction errors in such cases. However, it is observed that simply encoding temporal information is not sufficient for effective video anomaly detection. Therefore, the trajectory-based anomaly detection approach is integrated with state-of-the-art methods that rely on spatial cues. The network diagram illustrating the proposed approach is depicted in Figure 5.4. The effectiveness of integrating the trajectory-based anomaly detection approach with the state-of-the-art method is demonstrated through the results presented in Table 5.5. The table illustrates that the proposed system surpasses the baseline method's performance by incorporating temporal cues from trajectories.

4 Summary

This chapter focuses on the development of self-supervised tasks tailored to address anomaly detection problem. Three distinct approaches are proposed, each targeting the anomaly detection problem from a different perspective. The first two methods leverage the context provided by predicting masked pixels, which is essential for effective anomaly detection. In contrast, the third approach incorporates lightweight trajectory features to enhance unsupervised anomaly detection by incorporating long-range motion information. The contributions considered in this chapter are as follows:

- Proposes a novel self-supervised predictive convolutional block (SSPCAB). SSPCAB's components are 2D masked convolution, channel attention using Squeeze-and-Excitation (SE) module [119], and MSE as a self-supervised loss. This block possesses inherent anomaly detection capabilities.

- SSPCAB is integrated with four video anomaly detection baselines and two

Method	Avenue				ShanghaiTech			
	AUC		RBDC	TBDC	AUC		RBDC	TBDC
	Micro	Macro			Micro	Macro		
Liu *et al.* [16]	85.1	81.7	19.59	56.01	72.8	80.6	17.03	54.23
Liu *et al.* [16] + SSPCAB [24]	87.3	84.5	20.13	62.30	74.5	82.9	18.51	60.22
Liu *et al.* [16] + SSMCTB [69]	**89.5**	**84.6**	**23.79**	**66.03**	**74.6**	**83.9**	**19.13**	**61.65**
He *et al.* [103]	84.0	85.6	-	-	74.3	81.1	-	-
He *et al.* [103] + SSPCAB [24]	85.1	85.8	-	-	74.5	**81.9**	-	-
He *et al.* [103] + SSMCTB [69]	**86.4**	**86.5**	-	-	**76.1**	81.6	-	-
Park *et al.* [19]	82.8	86.8	-	-	68.3	79.7	-	-
Park *et al.* [19] + SSPCAB [24]	84.8	**88.6**	-	-	69.8	80.2	-	-
Park *et al.* [19] + SSMCTB [69]	**87.0**	87.7	-	-	**70.6**	**80.3**	-	-
Liu *et al.* [64]	89.9	93.5	41.05	86.18	74.2	83.2	44.41	83.86
Liu *et al.* [64] + SSPCAB [24]	**90.9**	92.2	**62.27**	**89.28**	**75.5**	83.7	45.45	84.50
Liu *et al.* [64] + SSMCTB [69]	89.6	**93.9**	46.49	86.43	75.2	**83.8**	**45.86**	**84.69**
Georgescu *et al.* [65]	92.3	90.4	65.05	**66.85**	82.7	89.3	**41.34**	78.79
Georgescu *et al.* [65] + SSPCAB [24]	92.9	**91.9**	65.99	64.91	**83.6**	**89.5**	40.55	**83.46**
Georgescu *et al.* [65] + SSMCTB [69]	**93.2**	91.8	**66.04**	65.12	83.3	**89.5**	40.52	81.93
Bărbălău *et al.* [79]	91.6	**92.5**	47.83	85.26	**83.8**	90.5	47.14	85.61
Bărbălău *et al.* [79] + 3D SSMCTB [69]	91.6	92.4	**49.01**	**85.94**	83.7	**90.6**	**47.73**	**85.68**

Table 5.4: Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech. Among the existing models, we select six models [16, 19, 64, 65, 79, 103] to show results before and after including SSPCAB and SSMCTB, respectively. The best result for each underlying model is highlighted in bold. The top score for each metric is shown in red. [Table and caption are taken from [69]]

Method	AUC(↑)	AUC(↑)
	Avenue	SH-Tech
Ours (Temporal Only: SGAN)	65.0	69.7
Spatial Only: Liu et. al. [16]	85.1	72.8
Ours (Spatial: Liu et. al., Temporal: SGAN)	**86.8**	**74.6**
Spatial Only: Park et. al. - Pred [19]	88.5	70.5
Ours (Spatial: Park et. al. - Pred., Temporal: SGAN)	**88.6**	**73.8**
Spatial Only: Park et. al. - Reconst [19]	82.8	69.8
Ours (Spatial: Park et. al. - Reconst., Temporal: SGAN)	**86.9**	**73.2**

Table 5.5: Comparing the frame-level AUC score (in %) of the proposed system with the SoTA approaches and their corresponding spatial anomaly detection branch. Higher frame-level AUC indicates better performance. [Table and caption are taken from [23]]

image anomaly detection baselines. This results in SOTA performance on unsupervised video anomaly detection tasks.

- Proposes a self-supervised masked convolutional transformer block (SSMCTB). The components of SSMCTB are 2D/3D masked convolution, channel attention using transformers, and Huber as a self-supervised loss. This block further improves over SSPCAB.

- A dataset for anomaly detection in the thermal domain is proposed, generated by annotating videos from one week of data from the Season-in-drift benchmark [120].
- SSMCTB is applied to datasets from various domains, including RGB, thermal, and medical for image and video anomaly detection tasks. This demonstrates its effectiveness across different data modalities.
- A method proposes to incorporate long-range temporal cues from trajectories. The approach considers motion and interaction characteristics of trajectories as an underlying feature for anomaly detection.
- The temporal cues obtained from trajectories are integrated with spatial cues derived from frame embeddings (in SOTA approaches), resulting in improved performance over baseline methods.

While this thesis focuses on the specific application of unsupervised learning for anomaly detection, the concept of self-supervised tasks has broader applicability to various computer vision tasks.

References

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," in *Proceedings of CVPR*, 2019, pp. 9592–9600.

[2] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, "Defect Detection in SEM Images of Nanofibrous Materials," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 551–561, 2017.

[3] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proceedings of ICPR*, 2021, pp. 475–489.

[4] Y. Fei, C. Huang, C. Jinkun, M. Li, Y. Zhang, and C. Lu, "Attribute Restoration Framework for Anomaly Detection," *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[5] Z. Li, N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, "Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection," in *Proceedings of BMVC*, 2020.

[6] M. Rudolph, B. Wandt, and B. Rosenhahn, "Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows," in *Proceedings of WACV*, 2021, pp. 1907–1916.

[7] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, "Multiresolution Knowledge Distillation for Anomaly Detection," in *Proceedings of CVPR*, 2021, pp. 14 902–14 912.

[8] J. Yi and S. Yoon, “Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation,” in *Proceedings of ACCV*, 2020, pp. 375–390.

[9] W. Sultani, C. Chen, and M. Shah, “Real-world anomaly detection in surveillance videos,” in *Proceedings of the CVPR*, 2018, pp. 6479–6488.

[10] Y. Tian, G. Pang, Y. Chen, R. Singh, J. W. Verjans, and G. Carneiro, “Weakly-supervised video anomaly detection with robust temporal feature magnitude learning,” in *Proceedings of the ICCV*, 2021, pp. 4955–4966.

[11] B. Wan, Y. Fang, X. Xia, and J. Mei, “Weakly supervised video anomaly detection via center-guided discriminative learning,” in *Proceedings of International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[12] C. Qiu, A. Li, M. Kloft, M. R. Rudolph, and S. Mandt, “Latent outlier exposure for anomaly detection with contaminated data,” in *Proceedings of the International Conference on Machine Learning(ICML)*, 2022.

[13] M. Zaheer, A. Mahmood, M. Astrid, and S.-I. Lee, “Claws: Clustering assisted weakly supervised learning with normalcy suppression for anomalous event detection,” in *Proceedings of the ECCV*, 2020.

[14] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, “Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection,” in *Proceedings of ICCV*, 2019, pp. 1705–1714.

[15] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in *Proceedings of CVPR*, 2016, pp. 733–742.

[16] W. Liu, W. Luo, D. Lian, and S. Gao, “Future Frame Prediction for Anomaly Detection – A New Baseline,” in *Proceedings of CVPR*, 2018, pp. 6536–6545.

[17] W. Luo, W. Liu, and S. Gao, “A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework,” in *Proceedings of ICCV*, 2017, pp. 341–349.

[18] T.-N. Nguyen and J. Meunier, “Anomaly Detection in Video Sequence With Appearance-Motion Correspondence,” in *Proceedings of ICCV*, 2019, pp. 1273–1283.

[19] H. Park, J. Noh, and B. Ham, “Learning Memory-guided Normality for Anomaly Detection,” in *Proceedings of CVPR*, 2020, pp. 14 372–14 381.

[20] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, “Abnormal Event Detection in Videos using Generative Adversarial Nets,” in *Proceedings of ICIP*, 2017, pp. 1577–1581.

[21] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video,” in *Proceedings of CVPR*, 2019, pp. 7842–7851.

[22] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “Anomaly Detection in Video via Self-Supervised and Multi-Task Learning,” in *Proceedings of CVPR*, 2021, pp. 12 742–12 752.

[23] N. Madan, A. Farkhondeh, K. Nasrollahi, S. Escalera, and T. B. Moeslund, “Temporal cues from socially unacceptable trajectories for anomaly detection,” in *Proceedings of ICCV-W*, October 2021, pp. 2150–2158.

[24] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection," in *Proceedings of CVPR*, 2022, pp. 13 576–13 586.

[25] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognition Letters*, vol. 129, pp. 123–130, 2020.

[26] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *Proceedings of ECCV*, 2020, pp. 485–503.

[27] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRAEM – A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection," in *Proceedings of ICCV*, 2021, pp. 8330–8339.

[28] S. Yamada, S. Kamiya, and K. Hotta, "Reconstructed Student-Teacher and Discriminative Networks for Anomaly Detection," in *Proceedings of IROS*, 2022, pp. 2725–2732.

[29] M. Haselmann, D. P. Gruber, and P. Tabatabai, "Anomaly detection using deep learning based image completion," *Proceedings of ICMLA*, pp. 1237–1242, 2018.

[30] J. Jiang, J. Zhu, M. Bilal, Y. Cui, N. Kumar, R. Dou, F. Su, and X. Xu, "Masked Swin Transformer Unet for Industrial Anomaly Detection," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2200–2209, 2023.

[31] J. Pirnay and K. Chai, "Inpainting transformer for anomaly detection," in *Proceedings of ICIAP*, 2022, pp. 394–406.

[32] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, "Cloze Test Helps: Effective Video Anomaly Detection via Learning to Complete Video Events," in *Proceedings of ACMMM*, 2020, pp. 583–591.

[33] M. Sabokrou, M. PourReza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, "AVID: Adversarial Visual Irregularity Detection," in *Proceedings of ACCV*, 2018, pp. 488–505.

[34] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proceedings of CVPR*, 2015, pp. 2909–2917.

[35] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of CVPR*, 2011, pp. 3449–3456.

[36] J. K. Dutta and B. Banerjee, "Online Detection of Abnormal Events Using Incremental Coding Length," in *Proceedings of AAAI*, 2015, pp. 3755–3761.

[37] C. Lu, J. Shi, and J. Jia, "Abnormal Event Detection at 150 FPS in MATLAB," in *Proceedings of ICCV*, 2013, pp. 2720–2727.

[38] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund, "Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection," in *Proceedings of BMVC*, 2015, pp. 28.1–28.13.

[39] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings," in *Proceedings of CVPR*, 2020, pp. 4183–4192.

[40] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, “Detecting abnormal events in video using Narrowed Normality Clusters,” in *Proceedings of WACV*, 2019, pp. 1951–1960.

[41] B. Ramachandra and M. Jones, “Street Scene: A new dataset and evaluation protocol for video anomaly detection,” in *Proceedings of WACV*, 2020, pp. 2569–2578.

[42] B. Ramachandra, M. Jones, and R. Vatsavai, “Learning a distance function with a Siamese network to localize anomalies in videos,” in *Proceedings of WACV*, 2020, pp. 2598–2607.

[43] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, “Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection,” in *Proceedings of WACV*, 2018, pp. 1689–1698.

[44] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[45] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, “Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,” *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[46] V. Saligrama and Z. Chen, “Video anomaly detection based on local statistical aggregates,” in *Proceedings of CVPR*, 2012, pp. 2112–2119.

[47] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, “Deep Appearance Features for Abnormal Behavior Detection in Video,” in *Proceedings of ICIAP*, vol. 10485, 2017, pp. 779–789.

[48] Q. Sun, H. Liu, and T. Harada, “Online growing neural gas for anomaly detection in changing surveillance scenes,” *Pattern Recognition*, vol. 64, no. C, pp. 187–201, Apr. 2017.

[49] H. T. Tran and D. Hogg, “Anomaly Detection using a Convolutional Winner-Take-All Autoencoder,” in *Proceedings of BMVC*, 2017.

[50] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. V. Gehler, “Towards Total Recall in Industrial Anomaly Detection,” in *Proceedings of CVPR*, 2022, pp. 14 298–14 308.

[51] A. Del Giorno, J. Bagnell, and M. Hebert, “A Discriminative Framework for Anomaly Detection in Large Videos,” in *Proceedings of ECCV*, 2016, pp. 334–349.

[52] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, “Unmasking the abnormal events in video,” in *Proceedings of ICCV*, 2017, pp. 2895–2903.

[53] Y. Liu, C.-L. Li, and B. Póczos, “Classifier Two-Sample Test for Video Anomaly Detections,” in *Proceedings of BMVC*, 2018.

[54] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, “Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection,” in *Proceedings of CVPR*, 2020, pp. 12 173–12 182.

[55] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[56] B. Antic and B. Ommer, “Video parsing for abnormality detection,” in *Proceedings of ICCV*, 2011, pp. 2415–2422.

[57] Y. Feng, Y. Yuan, and X. Lu, “Learning deep event models for crowd anomaly detection,” *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[58] R. Hinami, T. Mei, and S. Satoh, “Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge,” in *Proceedings of ICCV*, 2017, pp. 3639–3647.

[59] J. Kim and K. Grauman, “Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates,” in *Proceedings of CVPR*, 2009, pp. 2921–2928.

[60] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, “Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 1975–1981.

[61] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in *Proceedings of CVPR*, 2009, pp. 935–942.

[62] B. Saleh, A. Farhadi, and A. Elgammal, “Object-Centric Anomaly Detection by Attribute-Based Reasoning,” in *Proceedings of CVPR*, 2013, pp. 787–794.

[63] S. Wu, B. E. Moore, and M. Shah, “Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 2054–2060.

[64] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, “A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction,” in *Proceedings of ICCV*, 2021, pp. 13 588–13 597.

[65] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[66] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, “Old is gold: Redefining the adversarially learned one-class classifier training paradigm,” in *Proceedings of the CVPR*, 2020, pp. 14 183–14 193.

[67] M. Astrid, M. Z. Zaheer, and S.-I. Lee, “Synthetic Temporal Anomaly Guided End-to-End Video Anomaly Detection,” in *Proceedings of ICCVW*, 2021, pp. 207–214.

[68] H. M. Schlüter, J. Tan, B. Hou, and B. Kainz, “Natural synthetic anomalies for self-supervised anomaly detection and localization,” in *Proceedings of ECCV*, 2022.

[69] N. Madan, N.-C. Ristea, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, “Self-supervised masked convolutional transformer block for anomaly detection,” *arXiv*, 2022.

[70] 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 *Proceedings of NIPS*, vol. 30, 2017.

[71] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[72] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, vol. 37, no. 9, pp. 1904–1916, 2015.

[73] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," in *Proceedings of BMVC*, 2018.

[74] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the ECCV*, 2018, p. 3–19.

[75] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[76] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[77] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[78] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the CVPR*, 2015, pp. 3431–3440.

[79] A. Bărbălău, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "SSMTL++: Revisiting Self-Supervised Multi-Task Learning for Video Anomaly Detection," *Computer Vision and Image Understanding*, vol. 229, p. 103656, 2023.

[80] M. Astrid, M. Z. Zaheer, J.-Y. Lee, and S.-I. Lee, "Learning not to reconstruct anomalies," in *Proceedings of BMVC*, 2021.

[81] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of ECCV*, 2020, pp. 213–229.

[82] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation," *arXiv preprint arXiv:2102.04306*, 2021.

[83] 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," in *Proceedings of ICLR*, 2021.

[84] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Computing Surveys*, 2021.

[85] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," in *Proceedings of ICML*, 2018, pp. 4055–4064.

[86] N.-C. Ristea, A.-I. Miron, O. Savencu, M.-I. Georgescu, N. Verga, F. S. Khan, and R. T. Ionescu, "CyTran: Cycle-Consistent Transformers for Non-Contrast to Contrast CT Translation," *arXiv preprint arXiv:2110.06400*, 2021.

[87] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of ICML*, 2021, pp. 10 347–10 357.

[88] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," in *Proceedings of ICCV*, 2021, pp. 22–31.

[89] X. Xu and N. Xu, "Hierarchical Image Generation via Transformer-Based Sequential Patch Selection," in *Proceedings of AAAI*, 2022, pp. 2938–2945.

[90] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, “StyleSwin: Transformer-based GAN for High-resolution Image Generation,” in *Proceedings of CVPR*, 2022, pp. 11 304–11 314.

[91] M. Zheng, P. Gao, X. Wang, H. Li, and H. Dong, “End-to-end object detection with adaptive clustering transformer,” in *Proceedings of BMVC*, 2020.

[92] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable Transformers for End-to-End Object Detection,” in *Proceedings of ICLR*, 2020.

[93] Y. Lee and P. Kang, “AnoViT: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder,” *IEEE Access*, vol. 10, pp. 46 717–46 724, 2022.

[94] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, “VT-ADL: A vision transformer network for image anomaly detection and localization,” in *Proceedings of ISIE*. IEEE, 2021, pp. 1–6.

[95] G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, and D. Huang, “Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles,” in *Proceedings of the ECCV*, 2022.

[96] A. Barbalau, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, “Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection,” *Computer Vision and Image Understanding*, vol. 229, p. 103656, 2023.

[97] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[98] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in *Proceedings of the International Conference on Machine Learning (ICML)*, vol. 37, 2015, pp. 2342–2350.

[99] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *proceedings of the CVPR*, 2017, pp. 6299–6308.

[100] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the CVPR*, 2015, pp. 4489–4497.

[101] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, “Not only look, but also listen: Learning multimodal violence detection under weak supervision,” in *Proceedings of the ECCV*, 2020.

[102] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proceedings of NIPS*, 2014, pp. 2672–2680.

[103] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” in *Proceedings of CVPR*, 2022, pp. 16 000–16 009.

[104] 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 ICCV*, 2021, pp. 10 012–10 022.

[105] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, “Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning,” in *Proceedings of AAAI*, 2020, pp. 11 701–11 708.

[106] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, “Efficient anomaly monitoring over moving object trajectory streams,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, p. 159–168.

[107] C. CHEN, D. Zhang, P. S. CASTRO, N. Li, L. Sun, S. LI, and Z. WANG, “iBOAT : isolation-based online anomalous trajectory detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806–818, 2013.

[108] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato, and Yozo Fujino, “Learning motion patterns and anomaly detection by human trajectory analysis,” in *2007 IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 498–503.

[109] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, *et al.*, “Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications,” in *Proceedings of the world wide web conference*, 2018, pp. 187–196.

[110] E. Keogh, J. Lin, and A. Fu, “Hot sax: Efficiently finding the most unusual time series subsequence,” in *Proceedings of International Conference on Data Mining (ICDM)*. Ieee, 2005, pp. 8–pp.

[111] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, “A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data,” in *Proceedings of the AAAI*, vol. 33, no. 01, 2019, pp. 1409–1416.

[112] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, “Usad: Unsupervised anomaly detection on multivariate time series,” in *Proceedings of the International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3395–3404.

[113] Y. Ji, L. Wang, W. Wu, H. Shao, and Y. Feng, “A method for lstm-based trajectory modeling and abnormal trajectory detection,” *IEEE Access*, vol. 8, pp. 104 063–104 073, 2020.

[114] B. Kim, C. M. Kang, J. Kim, S.-H. Lee, C. C. Chung, and J. W. Choi, “Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network,” in *Proceedings of International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 399–404.

[115] G. Bouritsas, S. Daveas, A. Danelakis, and S. C. A. Thomopoulos, “Automated realtime anomaly detection in human trajectories using sequence to sequence networks,” in *Proceedings of Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–8.

[116] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” in *Proceedings of the CVPR*, 2016.

[117] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *Proceedings of the CVPR*, 2018.

[118] P. J. Huber, “Robust Estimation of a Location Parameter,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[119] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *Proceedings of CVPR*, 2018, pp. 7132–7141.

[120] 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 *Proceedings of NeurIPS*, 2021.

[121] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, “The Multimodal Brain Tumor Image Segmentation Benchmark (BraTS),” *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

Chapter 6

Conclusions and Future Work

This chapter presents the concluding remarks on different learning methodologies providing an alternative solution to mitigate the requirement of large-scale human-annotated datasets. The approaches include using synthetic datasets, representation learning, and unsupervised learning, which are presented in the order of labeled data requirements in this thesis. Additionally, the chapter contains possible future extensions of the research works discussed in this thesis.

1 Conclusions

The first chapter of this thesis addresses the problem of multi-object tracking by leveraging large-scale human-annotated datasets. While these datasets are readily available for well-researched problems like multi-object tracking, manually annotating extensive data becomes impractical for new visual tasks. Therefore, the subsequent chapters in this thesis provide alternative solutions for dealing with this issue. This section primarily contains concluding remarks on each proposed solution.

Synthetic Datasets. This thesis explores the specific problem of detecting people falling into water, which occurs infrequently, and hence the collection of datasets is a challenging task to solve the problem. To overcome this limitation, the thesis proposes synthetic datasets. Synthetic datasets offer the advantage of annotations readily available for free. The key consideration is ensuring that models trained on synthetic datasets exhibit transferability to real-world datasets.

In the proposed task of detecting people falling into the water, the synthetic dataset demonstrates exemplary transferability characteristics when applied to real-world data. Notably, synthetic datasets have shown promise in addressing various computer vision problems, as evidenced by prior research [1–6] including fundamental tasks such as object detection and segmentation. Nevertheless, synthetic datasets may not consistently exhibit effective transfer characteristics across all visual tasks. To enhance the transferability of synthetic datasets, some approaches [7, 8] combine synthetic and real datasets and are often employed to tackle complex tasks. These approaches results in diverse datasets, leading to improved generalization. In this thesis, the proposed dataset is also generated by combining components from real and synthetic world data, aiming to enhance transferability.

Representation Learning. This thesis presents an approach to representation learning using masked image modeling (MIM) as *pretext task*. Inspired by the success and simplicity of MAE [9], the proposed approach employ MAE [9] as the representation learning backbone. While current SOTA [9–12] approaches randomly mask tokens to solve the MIM. This thesis presents an end-to-end adaptively learnable masking strategy, which provides different making tokens as outputs based on task complexity. A curriculum [13] is finally designed based on task complexity to improve learned representation.

We present preliminary results in this thesis verifying that curriculum learning improves downstream task representation and hence performance under zero-shot settings. In our zero-short results, we do not use any labeled data, just the representation. Future work will extend the idea by generalizing it for multiple visual tasks.

Self-supervised representation learning provides a promising path, especially in surveillance data, where unlabelled data is substantial. However, the only constraint is that this large-scale dataset is not publicly available for annotation. One possible solution, in this case, might include using pre-trained models on large-scale out-of-domain image datasets and fine-tuning them under limited label settings. Another solution is to train in-house models on the large-scale surveillance dataset and use the learned representation to solve multiple visual tasks. The only constraint while pretraining these models is their high GPU memory requirement.

Unsupervised Learning. In conclusion, chapter 5 of this thesis introduces three approaches for addressing the anomaly detection problem using unsupervised learning. The first two approaches employ reconstruction as a mask modelling problem to enhance anomaly detection. The approaches propose two neural blocks namely SSPCAB and SSMCTB that can be integrated with any architecture. These blocks dynamically encode contextual information by configuring the mask convolution layer's receptive field and target pixels. Integration of these self-supervised blocks (SSPCAB and SSMCTB) with various baselines demonstrates significant performance improvements, surpassing state-of-the-art methods on multiple evaluation metrics. The results consistently indicate that the SSMCTB enhances anomaly detection performance across different domains (RGB, medical, thermal), highlighting the efficacy of modelling spatial and spatio-temporal context through configurable mask convolution.

The third approach combines long-range trajectories as additional temporal cues with state-of-the-art anomaly detection methods. This approach utilizes the Social-GAN framework [14] for detecting anomalous trajectories by capturing social interactions and past motion behavior. The results suggest the effectiveness of this approach, particularly for detecting group anomalies where abnormal social interactions, such as snatching or fighting, differ from normal patterns.

Overall, unsupervised learning eliminates the reliance on labeled data and offers significant advantages. However, the challenge lies in carefully designing self-supervised tasks to capture the high-level semantics of images and videos. In the case of anomaly detection, reconstruction proves to be a fitting self-supervised task, although its specific implementation varies depending on the visual task being addressed. As a result, task-specific representations are acquired, necessitating separate model training for each visual task.

2 Future Work

This subsection presents the possible extension of research based on the experience gained during this PhD.

Anomaly Detection. The problem of unsupervised anomaly detection has large similarities with other problem statements in computer vision such as novelty detection, out-of-distribution detection, and open-set recognition. Out of which, the problem of novelty detection matches exactly with anomaly detection. However, instead of novel classes we consider abnormal classes in anomaly detection. Similarly, out-of-distribution detection and open-set recognition have similarities in the regard that the training dataset is classified into K-classes instead of a single normal class and everything outside K-classes is treated as anomalies. A possible future extension of this thesis is the unification of all these different areas of research. A possible extension idea is to apply K-classes to anomaly detection. The concept will then extend to categorizing a single normal class into multiple categories. This categorization can be performed based on actions in videos for instance. The idea can also be further extended to anomalies categorization. To achieve this categorization, we can transform the problem of generic category discovery [15], where we cluster the novel action categories found in the abnormal test samples seen at test time. Obtaining these categorizations for anomalies is especially interesting because each anomaly has a different implication, some of which are more critical than others.

Solving Dataset Biases. The ultimate objective of the learning process is to develop models that demonstrate strong generalization capabilities in real-world scenarios. However, achieving this goal is a complex task due to biases in publicly available datasets. These biases, whether acknowledged or unidentified, present significant challenges for models to generalize to real-world datasets. We discuss some solutions for solving bias that comes as noise, during the manual data-annotation process in this thesis.

However, we observed another bias during our investigation of the publicly available video anomaly detection dataset. The bias, in this case, is related to motion-based anomalies, which limit the inclusion of complex spatio-temporal features. As a result, these datasets heavily rely on optical flow as the primary source of temporal information. This restricts their ability to generalize when applied to real-world surveillance cameras that encompass more complex interaction-related anomalies beyond mere motion-based ones like running or jumping.

In addition to the acknowledged biases, various undetected biases exist across different computer vision tasks, including object detection, semantic segmentation, multi-object tracking, and anomaly detection. To mitigate the impact of biases, several proposed solutions [16–20] involve training models on large-scale and diverse datasets. By incorporating a wide range of data from various sources, models have a better chance of learning robust and generalized representations that are not biased toward specific characteristics present in limited datasets.

Following a similar trend, an idea for addressing this issue could involve expanding existing datasets in the anomaly detection domain. This could be done by incorporating more diverse scenarios. One approach to add diversity is by augmenting datasets with synthetic anomalies. The rationale behind adding synthetic anomalies is that the rarity of anomalies makes it challenging to capture every instance in the dataset. Augmenting the dataset with synthetically generated anomalies provides flexibility in including more complex spatio-temporal anomalies encountered in the real world. Furthermore, augmenting datasets with synthetic objects or humans will meet the privacy-preserving requirements of the GPPR [21] rules.

In conclusion, the learning process aims to develop models that can effectively generalize across various real-world scenarios. Biases, whether introduced during the learning process or present in the dataset, hinder the model's generalization capabilities. The research in this thesis addresses human biases originating from labeled datasets. The proposed solutions involve augmenting datasets with synthetic components, and rectifying identified biases. Additionally, leveraging a smaller amount of labeled data or just the unlabeled data proves beneficial in mitigating human biases inherent in the labeling process.

References

[1] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, "Motsynth: How can synthetic data help pedestrian detection and tracking?" in *Proceedings of ICCV*, 2021, pp. 10 849–10 859.

[2] C. Pramerdorfer, J. Strohmayer, and M. Kampel, "Sdt: A synthetic multi-modal dataset for person detection and pose classification," in *Proceedings of ICIP*, 2020, pp. 1611–1615.

[3] J. Marín, D. Vázquez, D. Gerónimo, and A. M. López, "Learning appearance in virtual scenarios for pedestrian detection," in *Proceedings of CVPR*, 2010, pp. 137–144.

[4] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, “Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 746–753.

[5] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of CVPR*, 2020, pp. 2443–2451.

[6] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *Proceedings of the CVPR*, June 2016.

[7] N. Madan, M. S. N. Siemon, M. K. Gjerde, B. S. Petersson, A. Grotuzas, M. A. Esbensen, I. A. Nikolov, M. P. Philipsen, K. Nasrollahi, and T. B. Moeslund, “Thermalsynth: A novel approach for generating synthetic thermal human scenarios,” in *Proceedings of the WACVW*, 2023, pp. 130–139.

[8] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz, “Context-aware synthesis and placement of object instances,” *Advances in neural information processing systems*, 2018.

[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” in *Proceedings of the CVPR*, 2022, pp. 16 000–16 009.

[10] H. Bao, L. Dong, S. Piao, and F. Wei, “BEit: BERT pre-training of image transformers,” in *Proceeding of ICLR*, 2022.

[11] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, “Simmim: A simple framework for masked image modeling,” in *Proceedings of the CVPR*, June 2022, pp. 9653–9663.

[12] Y. Shi, N. Siddharth, P. H. Torr, and A. R. Kosiorek, “Adversarial masking for self-supervised learning,” in *Proceeding of the International Conference on Machine Learning (ICML)*, 2022.

[13] Y. Bengio, “Learning deep architectures for AI,” *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[14] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in *Proceedings of the CVPR*, 2018.

[15] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, “Open-set recognition: A good closed-set classifier is all you need,” in *Proceedings of the ICLR*, 2022.

[16] S. Madan, T. Henry, J. Dozier, H. Ho, N. Bhandari, T. Sasaki, F. Durand, H. Pfister, and X. Boix, “When and how convolutional neural networks generalize to out-of-distribution category–viewpoint combinations,” *Nature Machine Intelligence*, vol. 4, no. 2, pp. 146–153, 2022.

[17] S. Sinha, H. Bharadhwaj, A. Goyal, H. Larochelle, A. Garg, and F. Shkurti, “Dibs: Diversity inducing information bottleneck in model ensembles,” in *Proceedings of the AAAI*, 2020.

[18] A. Cui, A. Sadat, S. Casas, R. Liao, and R. Urtasun, “Lookout: Diverse multi-future prediction and planning for self-driving,” in *Proceedings of the ICCV*, 2021, pp. 16 087–16 096.

[19] Y. Yang, A. Gupta, J. Feng, P. Singhal, V. Yadav, Y. Wu, P. Natarajan, V. Hedau, and J. Joo, “Enhancing fairness in face detection in computer vision systems by demographic bias mitigation,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2022.

[20] A. Deviyani, “Assessing dataset bias in computer vision,” *ArXiv*, vol. abs/2205.01811, 2022.

[21] A. v. d. B. Paul Voigt, *The EU General Data Protection Regulation (GDPR) - A practical guide*. Cham: Springer International Publishing, 2017.

Part II

Papers

Paper A

Attention-Enabled Object Detection to Improve One-Stage Tracker

Neelu Madan, Kamal Nasrollahi, and Thomas B. Moeslund

The paper has been published in the
IntelliSys 2021: Intelligent Systems and Applications Vol. 295, p. 736–754, 2021.

Abstract

State-of-the-art (SoTA) detection-based tracking methods mostly accomplish the detection and the identification feature learning tasks separately. Only a few efforts include the joint learning of detection and identification features. This work proposes two novel one-stage trackers by introducing implicit and explicit attention to the tracking research topic. For our tracking system based on implicit attention, we further introduce a novel fusion of feature maps combining information from different abstraction levels. For our tracking system based on explicit attention, we introduce utilization of an additional auxiliary function. These systems outperform the SoTA tracking systems in terms of MOTP (Multi-Object Tracking Precision) and IDF1 score when evaluated on public benchmark datasets including MOT15, MOT16, and MOT17. High MOTP score indicates precise detection of bounding boxes of objects, while high IDF1 score indicates accurate ID detections, which is very crucial for surveillance and security systems. Therefore, proposed systems are good choice for event-detections in surveillance feeds as we are capable of detecting correct ID and precise location.

1 Introduction

Multi Object Tracking (MOT) is one of the most widely used and yet challenging applications of computer vision. The aim is to predict the object trajectories across the video frames. The predicted trajectories could further be used for the analysis of the sports videos [1], anomaly detection in crowded scenes [2], Automatic Driving Assistance Systems [3], to name a few.

The challenges related to visual object tracking such as occlusion, motion blur, social interaction, and low-resolution images make it a complicated task. Myriads of approaches are proposed overtime to resolve the multiple challenges. Some of the earliest efforts in this field used correlation-filter [4], and mean-shift algorithm [5]. Later on, deep learning-based tracking approaches become popular due to an increase in GPU computation power. Some of the widely used deep learning based tracking approaches include Siamese Neural Network (SNN) [6] which learns similarity function between target and search region, Recurrent Neural Networks (RNN) [7] which incorporate temporal features in addition to appearance based features, and Generative Adversarial Network (GAN) [8] based tracker which works well even if training dataset is small.

The success of deep neural networks in object detection paved the way for developing trackers using detectors in their backbone also know as tracking-by-detection. Some research in this area includes two-stage and one-stage trackers. In a two-stage trackers, target objects are detected as Bounding Boxes (BBoxes) by an object detector in the first stage. In the second stage, a unique identification (ID) is assigned to each BBox using a different network. Then, usually an affinity/cost matrix is constructed

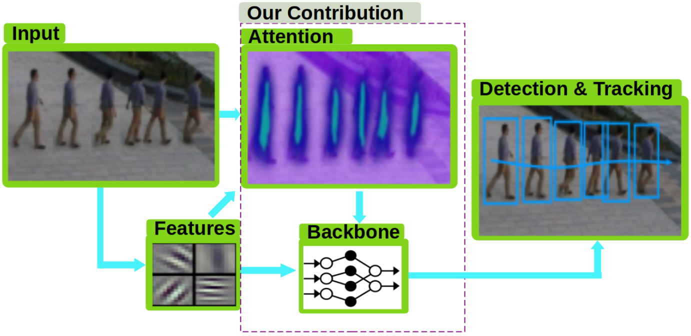


Fig. A.1: Illustrating the proposed tracking system.

by combining Intersection over Union (IoU) of detection BBox and the assigned ID. Once the affinity matrix is created, an algorithm like Hungarian [9] is used to associate the target object by minimizing the total cost function. Reliable tracking accuracy can be achieved by using the best detection and identification networks in a two-stage tracker. However, this also increases the training complexity of the network and its inference time. To reduce the training effort, one-stage trackers [10] performing both the detection and the identification task in a single network using multi-task learning are proposed. In multi-task learning [11], a shared network is used to learn multiple tasks, here detection and re-identification (re-id). However, this joint learning reduces the network's generalization capabilities by sharing the same low-level and high-level features for both tasks. Therefore, the accuracy of the one-stage trackers is lower as compared to their two-stage counterparts. On further analysis, the unstable detection at small scale seems to be one of the major causes of decreased accuracy of one-stage trackers. This paper improves the detection accuracy of one-stage trackers adding different attention modules to the backbone architecture. Additionally, the proposed work also improved the identity detection by enhancing the feature propagation.

The accuracy of detection-based one-stage trackers can be improved by either improving the detection task or association-task. State-of-The-Art (SoTA) approaches such as Chained-Tracker [12] improves the data-association by incorporating task-specific attentions. In this paper, we focused on improving the detection task and modified existing backbone architectures of a known network that has been used mostly for object detection, HRNet [13], by introducing different attention modules to propose a novel one-stage tracker as shown in Figure A.1. The concept of attention networks [14] is motivated by the human visual system, which learns to focus on a relevant region of the image while discarding the irrelevant part. In the past few years, attention-based networks have proved to be very successful in improving the performance of multiple computer vision tasks [15] and [16].

Attention mechanisms can be broadly classified into two categories, i.e., implicit [17] and explicit attention [18]. Implicit attention doesn't enforce any additional constraints on the attention unit. On the contrary, learning attention units explicitly with additional supervision from ground-truth (GT) enforces the network to improve on certain tasks. In this paper, we have introduced attention (in both forms of implicit [19] and explicit [20] attention) to one-stage trackers and shown that this improves the accuracy of the trackers.

The contributions of this paper are as follows:

1. We, for the first time, introduce attention, in its two forms of implicit and explicit attentions, to one-stage trackers based on multi-task learning, resulting in two one-stage trackers (one based on implicit attention and one based on explicit attention) which outperform SoTA trackers in terms of MOTP and IDF1 score on public benchmark datasets, while achieving competitive results on MOTA score. The proposed attention-gates improves the feature propagation across the later stages of the network and hence the ID detection.
2. For our tracker based on implicit attention, we propose a novel fusion of feature maps combining information coming from different abstraction levels, that improves the accuracy of the tracker even further (Section 5, Table A.4).
3. For our tracker based on explicit attention, we propose a novel modification to our employed backbone architecture via an auxiliary loss function, that further improves the tracking accuracy (Section 5, Table A.5).

The rest of this paper is organized as follows: The related work in the literature is reviewed in the next section. Then, Section 3 explains the details of the proposed idea. Section 4, gives the details of the experimental results comparing the performance of the system with the SoTA systems. Section 5 presents the ablation study, and finally the paper is concluded in Section 6.

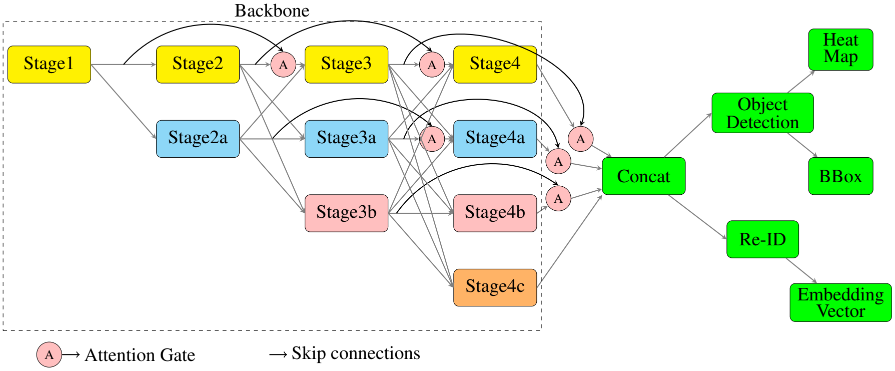


Fig. A.2: Introducing implicit attention to a one-stage tracker based on HRNet [13] backbone.

2 Related Work

This section is divided into two parts. First, we review SoTA MOT approaches focusing on detection-based trackers. Then, we review the reported incorporation of implicit and explicit attention for improving the performance of deep learning-based computer vision methods.

2.1 Deep Learning-based Tracking

Most of the recent research works in MOT are using deep-learning-based architectures. The recent major contributions include Graph Convolutional Network (GCN) [21], multi-task learning [22], articulated tracking [23], and tracking-by-detection [24]. SoTA GCN-based trackers consist of two different GCNs, i.e., spatial-temporal GCN represents the structure and contextual GCN models the context. Multi-task learning-based architectures jointly optimizing on different related tasks such as: Multi-Object Tracking and Segmentation (MOTS) [22] optimize detection, tracking and segmentation tasks, JDE [10] optimizing detection and re-id tasks, and FAMNet [25] which is based on joint optimization of feature extraction, affinity estimation, and multi-dimension assignment. Articulated trackers track the key points across the video frames, e.g., architecture proposed in [23] contains two networks SpatialNet and TemporalNet. SpatialNet detects the body parts and groups them in a single frame and TemporalNet converts those key points into trajectories. Tracking-by-detection uses detected BBox to compute the affinity matrix, e.g., POI [26] incorporated an accurate object detector based on Faster-RCNN for pedestrian tracking. CenterNet [27] and Chained Tracker [12] proposed an integrated system performing simultaneous detection and association using consecutive frames as input.

Large-scale image recognition challenges such as ImageNet [28] are bringing up accurate and efficient object detectors, which oriented the research direction towards detector-based trackers [24]. There are multiple tracking approaches based on different object detection architecture such as YOLO [29], Faster-RCNN [30], and SSD [31]. Different association techniques discussed further are used to establish the temporal relationship and predict the trajectories of target objects. Deep association is proposed in [32], where a separate deep neural network is used to accomplish the association task. A tubelet proposal network [33] incorporated temporal features using Long Short-Term Memory (LSTM) networks. Detection-based tracking approaches mentioned above are computationally expensive and suffer from increased inference time. To meet the real-time constraint, Tracktor++ [34] transformed object detector to a tracker by simply using regression and classification differently. This simple conversion improved the inference time but failed to improve accuracy due to the missing temporal relationships. To further improve the idea, one-stage trackers [10] which jointly learn detection for spatial and feature embedding for the temporal association are proposed.

2.2 Attention Gates

Attention gates have been initially used in natural language processing (NLP), e.g., hand-writing synthesis [35], and machine translation [36] to improve the contextual information in a text. More recently attention networks became a popular means to improve various computer-vision tasks such as image classification [16], and segmentation [37]. Not every feature generated by a deep neural network is equally important. Therefore, channel attention providing weights to different output channels has been applied in such cases. For example, squeeze-and-excitation networks proposed in [38] learn the channel weights by squeezing the spatial dimension. Additionally, few locations in the feature maps are more relevant than the others. Therefore, spatial attention [39] generates the attention map by utilizing the inter-spatial relationship of features. Some networks such as Convolutional Block Attention Module (CBAM) [40] have incorporated both channel and spatial attention in a single network.

Attention maps can be learned implicitly using self-guidance from the network itself or explicitly using external supervision. Guided inference network [20] learns attention explicitly by adding an auxiliary loss function. In general, it has been observed that the attention generated by additional supervision usually performs better as also mentioned in [20]. To the best of our knowledge, attention mechanisms either implicit or explicit have not previously been used for the accuracy improvement of one-stage tracker.

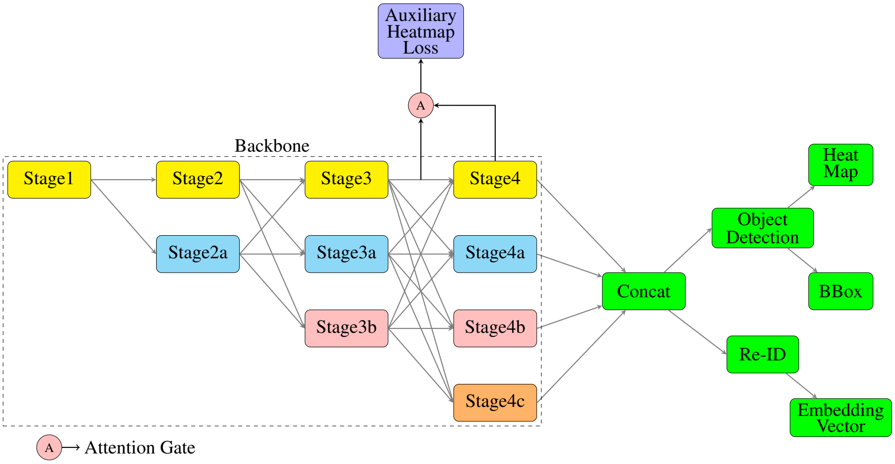


Fig. A.3: Introducing explicit attention to a one-stage tracker based on HRNet [13] backbone.

3 The Proposed Idea

This section discusses the details of the proposed idea of introducing attention to one-stage trackers, resulting in two systems, one using implicit and the other one explicit attention. This section also includes the details about similarities and differences between proposed explicit and implicit attention-based networks. We have introduced our attention idea to the HRNet [13] architecture as it maintains high-resolution throughout the network unlike Feature Pyramid Net (FPN) [29] based architecture, which consists of an encoder branch to reduce the resolution of feature-maps from high to low and a decoder branch to recover the high-resolution information. This property of HRNet [13] provided us with the scope for improving the detection accuracy for small objects by adding different attention modules. The following subsections present the proposed building blocks, network architectures, objective functions, and inference details of the two systems based on the different mentioned attentions.

3.1 Components

The basic building blocks for both implicit and explicit attention networks are the same. This section contains the details for the common blocks of the two networks as shown in Figure A.2 (implicit-based) and Figure A.3 (explicit-based).

Attention-gate

A basic function of any gate is to allow some information to pass through it while blocking the rest. The attention gates used in this paper are similar to the one proposed in [41]. The idea of the proposed attention gates is to filter the irrelevant features and prevent them from propagating across the network, which in turn improves the gradient flow of the network.

Backbone

HRNet [13] typically contains four stages, where a lower resolution is added to the network at the end of each stage. The basic structure of each stage remains unchanged as in original HRNet [13], i.e., each stage contains parallel multi-resolution convolution followed by multi-resolution fusion. For example, Stage 2 of HRNet [13] is shown in Figure A.4, and other stages are designed in a similar way. This repeated addition of new resolution and multi-resolution fusion increases the amount of global information after each stage [13]. The multi-resolution feature maps are concatenated and provided to the predictions heads, i.e., object detection and BBox estimation, as shown in Figure A.2 and A.3. These figures are based on generic block diagram of one-stage trackers [10, 42].

Object Detection

The object detection branch predicts two different outputs, i.e., heatmap and BBoxes' size and center offset. These are explained here:

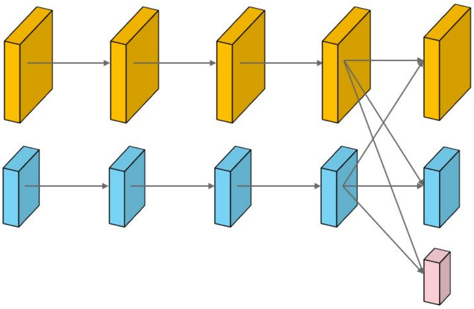


Fig. A.4: Stage 2 in the backbone of proposed architecture contains parallel convolution for two different resolution(yellow and blue) followed by the addition of new resolution (pink) and multi-resolution fusion represented by cross-connections.

Heatmap Estimation Heatmaps are mostly used in the context of key-point estimation [43]. We employ them to get the estimated position of the object's center and the most probable detection areas. The size of the predicted heatmap is the same as the input image but it contains only one channel. The output response of the predicted heatmap is expected to be one at the object center and decays exponentially with increase in distance from the center.

BBox Estimation BBox estimation predicts the size and offset for the target objects. The size of a BBox corresponds to its height and width around the probable regions proposed by the predicted heatmaps and offset to the object's center. The precise localization of the object has a significant role in a tracking system. The reason is, re-id features are extracted on the basis of the object's center. Therefore, accuracy in locating object's center needs to be high in order to improve the tracking system.

Re-Identification

The re-id branch generates the embedding vector, which distinguishes among different target objects and helps in predicting their corresponding tracks. The re-id branch learns a metric such that instances of the same identity are close to each other, and instances of different identities are far apart. To achieve the same, a convolutional layer predicting a 1-dimensional embedding vector of size 128 is introduced, which shares the same features as used for object detection. The embedding feature vector, corresponding to an object with center at (x,y) and is extracted from the feature map, which is finally used for association across the subsequent frames.

3.2 Network Architectures

Until now, we have discussed the structure of all the basic components, which are the same in both explicit and implicit attention networks. However, there are still some basic differences between their architectures which are described in this sub-section.

Implicit Attention The architecture of the proposed implicit attention-based network is shown in Figure A.2. Higher resolution in the HRNet [13] goes through a series of convolutional layers and keeps on adding more global information after each stage. Due to this long series of convolutions, the feature-map towards the end of the network tends to lose the information about the local context. To recover this information, attention-gated skip connections between the consecutive stages are proposed in the current research work. The reason for fusing the different feature-maps between the consecutive stages is discussed along with some experiments in Section 5. This attention-gated fusion of features weighs the lower-level features from the previous stage to complement the global features at the current stage and finally combines the information. Such connections are repeated for each resolution in the proposed architecture. This type of attention is called implicit because the attention-gate automatically learns to weigh the local context based on the existing global context.

Explicit Attention The network architecture of the proposed explicit attention-based network is shown in Figure A.3. We have introduced an auxiliary heatmap loss to our proposed explicit attention type which provides additional supervision. In this attention-type, a heatmap is predicted from the attention-gated feature at an intermediate stage. The attention gates here filter the feature maps at an intermediate stage based on the global information. An auxiliary loss between the predicted and ground-truth (GT) heatmap is calculated using the attention-gated feature-maps and later added to the objective function which is used during the network's training. The choice of the intermediate stage to add an auxiliary loss is decided experimentally and discussed in Section 5.

3.3 Loss Functions

This section contains the details of the overall objective which is obtained by combining multiple loss functions. The implicit and explicit attentions combine three and four types of different losses, respectively. The first three losses in explicit architecture are the same as that for implicit and acquired from [42]. Further details of the objective function are mentioned below.

Heatmap Loss Object center (c_x^k, c_y^k) is computed as $c_x^k = \frac{x_1^k + x_2^k}{2}$, and $c_y^k = \frac{y_1^k + y_2^k}{2}$ for each GT BBox $b^k = (x_1^k, y_1^k, x_2^k, y_2^k)$. The location of center on the feature map is obtained by dividing the stride, i.e., $(\tilde{c}_x^k, \tilde{c}_y^k) = \left(\left\lfloor \frac{c_x^k}{4} \right\rfloor, \left\lfloor \frac{c_y^k}{4} \right\rfloor \right)$. Heatmap response at location (x,y) is computed by using Gaussian distribution ,i.e. , $H_{xy} = \sum_{k=1}^{N} \exp^{\frac{(x-\tilde{c}_x^k)^2 + (y-\tilde{c}_y^k)^2}{2\sigma_c^2}}$, which shows that response is decreasing with increase in distance from the center. The loss function used for the regression of heatmap is pixel-wise logistic regression with focal loss. It is represented via Equation A.1 [42]:

$$\mathcal{L}_{hm} = -\frac{1}{N}\sum_{xy}\begin{cases}(1-\hat{H_{xy}})^{\alpha}\log(\hat{H_{xy}}), & \text{if } H_{xy}=1;\\ (1-H_{xy})^{\beta}(\hat{H_{xy}})^{\alpha}\log(1-\hat{H_{xy}}) & \text{otherwise,}\end{cases} \tag{A.1}$$

Offset and Size Loss The predictions from the offset and size head is denoted as $C_p \in \mathcal{R}^{W\times H\times 2}$ and $S_p \in \mathcal{R}^{W\times H\times 2}$ respectively, the size of single GT BBox with coordinates as $(x_1^k, y_1^k, x_2^k, y_2^k)$ is computed as $S = (x_2^k - x_1^k, y_2^k - y_1^k)$ and center offset is calculated as C = $C/4$. The detection loss of a single BBox is the sum of deviations in size and center offset. Total loss is obtained by the summation of K detected boxes as represented by equation A.2 [42]:

$$\mathcal{L}_{bbox} = \sum_{k=1}^{K} |C^k - C_p^k| + |S^k - S_P^k| \tag{A.2}$$

Re-ID Loss The ID prediction is considered a classification task where the number of classes corresponds to the number of IDs in the dataset, i.e., objects with the same ID are treated as one class. To calculate the Re-ID loss for each BBox (B_k), the soft-max function is applied to the predicted embedding vector to get the class distribution $P(\mathcal{N})$. GT can be represented as one-hot vector $GT^k(\mathcal{N})$. The loss function used for this case is a cross-entropy loss for multi-class classification as shown in Equation A.3 [42].

$$\mathcal{L}_{id} = -\sum_{k=1}^{K}\sum_{n=1}^{\mathcal{N}} GT^k(\mathcal{N}) log(P(\mathcal{N})) \tag{A.3}$$

where $\mathcal{N}$ and K are total number of classes and detected BBoxes respectively.

Auxiliary Loss In case of explicit attention, a heatmap is predicted from intermediate feature-maps. An auxiliary loss is added to optimize the heatmap prediction from an intermediate layer. The calculation of this auxiliary loss is similar to that of Equation A.1.

Loss Balancing based on Uncertainty of Tasks The overall objective of the networks is a weighted sum of the above-mentioned losses. However, the manual tuning of those weights is computationally expensive and difficult. Therefore, we used **automatic loss balancing** as proposed in [11] which uses task uncertainty to weigh the different losses. The total loss therefore can be represented via Equation A.4 [42]. Weights $\mathcal{W}_{hm}, \mathcal{W}_{bbox}, \mathcal{W}_{id}$ are learned automatically as part of neural network learning process, as follows:

$$
\begin{aligned}
\mathcal{L}_{total}^{Implicit} = \frac{1}{2}\Bigg(& \frac{1}{e^{\mathcal{W}_{hm}}}\mathcal{L}_{hm} + \frac{1}{e^{\mathcal{W}_{bbox}}}\mathcal{L}_{bbox} + \frac{1}{e^{\mathcal{W}_{id}}}\mathcal{L}_{id} \\
& + s_{hm} + s_{bbox} + s_{id}\Bigg)
\end{aligned}
\tag{A.4}
$$

where $\mathcal{L}_{hm}$, $\mathcal{L}_{bbox}$, and $\mathcal{L}_{id}$ are heatmap, BBox, ID loss respectively and s_{hm}, s_{bbox}, and s_{id} are task dependent uncertainties. In case of explicit attention-based architecture, total loss will be modified by adding an auxiliary loss ($\mathcal{L}_{aux}$) balanced via weight ($\mathcal{W}_{aux}$) and having task-dependent uncertainty of s_{aux} as in Equation A.5:

$$
\begin{aligned}
\mathcal{L}_{total}^{Explicit} = \frac{1}{2}\Bigg(& \frac{1}{e^{\mathcal{W}_{hm}}}\mathcal{L}_{hm} + \frac{1}{e^{\mathcal{W}_{bbox}}}\mathcal{L}_{bbox} + \frac{1}{e^{\mathcal{W}_{id}}}\mathcal{L}_{id} \\
& + \frac{1}{e^{\mathcal{W}_{aux}}}\mathcal{L}_{aux} + s_{hm} + s_{bbox} + s_{id} + s_{aux}\Bigg)
\end{aligned}
\tag{A.5}
$$

3.4 Inference and Online Association

Giving an input image to either of the two systems illustrated in Figure A.2 and A.3, they generate an output that can be categorized into two parts, i.e., detection and re-id. The detection part is represented via heatmap, BBox size, and offset for the detection task. The re-id part is depicted by the embedding vector. A non-maximum suppression (NMS) is performed based on heatmap scores on top of the predicted heatmap, which provides the most probable detection locations. The locations with scores greater than a certain threshold are kept and the rest are discarded, which is finally followed by the estimation of BBox's size and offset. The embedding vectors corresponding to the detections are also extracted. The next step is to associate the detected BBox and identity embeddings across the subsequent video frames.

Association is based on a standard online tracking algorithm as also discussed in [10]. Tracklets are first initialized based on the appearance features extracted from the first video frames and added to the tracklet pool. In the subsequent frames, pairwise motion and appearance similarity is calculated between the observations and tracklets from the pool. Metrics used for the association of appearance and motion-based features are Cosine similarity and Mahalanobis distance respectively. Finally, the assignment problem is resolved by using the Hungarian algorithm [9]. The appearance features are updated using a weighted combination of BBox IoU and embedding vector. However, motion cues are updated by using Kalman Filter [44]. The observations which are not assigned to any of the existing tracklets from the pool are marked new. On the contrary, tracklets are marked as terminated if no observation is found for a few subsequent frames.

4 Experimental Results

This section includes information about the evaluation metrics, dataset, and training methodology. In addition, this section also discusses the qualitative analysis, as well as compares our results with SoTA methods which have reported their results on the same benchmark datasets used in our experiments.

4.1 Evaluation Metrics and Dataset

The proposed system developed in this paper are compared using Multi Object Tracking Accuracy (MOTA) [45], MOTP [45], IDF1 Score [46], and number of ID Switches from CLEAR metric [45] as described below:

- Multi-object tracking accuracy (MOTA): Computes the overall tracking accuracy from false positives, false negatives and identity switches [45].
- Multi-object tracking precision (MOTP): Depicts overall tracking precision in terms of BBox overlap between ground-truth and predicted location [45].
- Identification F1 (IDF1): Measures the extent that predicted identities confront the ground-truth [46].
- Identity switches (ID): Number of times the reported identity of a ground-truth track changes.

The system is evaluated on **MOT15 train**, **MOT17 test**, and **MOT16 test** data, where **MOT15 train** data is used as validation set in our experiments. We are using the private protocol of MOT 16 and 17 under which we are allowed to use additional training data. We have therefore collected a set of additional training datasets including ETH dataset [47], CityPersons (CP) dataset [48], CalTech (CT) dataset [49], MOT17 (M17) dataset [50], CUHK-SYSU (CS) dataset [51], and PRW dataset [52]. The same datasets are used in the training of JDE [10], tracker that we have compared our results against. The joint dataset can be divided into two categories, i.e., ETH and CP contain annotations for detection only while CT, MOT17, CS, and PRW contain annotations for both detection and ID. The overlapping videos between the testing and training dataset are removed from the training data.

Dataset	Tracker	MOTA↑	IDF1↑	IDs↓
MOT15 Train	JDE [10]	67.5	66.7	218
	Ours(Implicit)	73.2	73.5	**203**
	Ours(Explicit)	**73.8**	**75.9**	232

Table A.1: Comparing the proposed trackers with the existing one-stage tracking approaches on MOT15 train dataset.

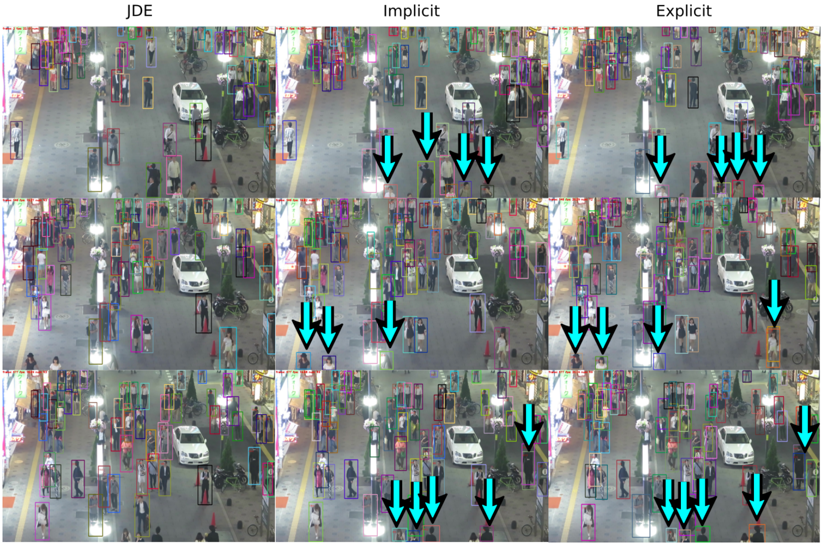


Fig. A.5: Comparison among the tracklets detected and tracked in the proposed systems (Explicit and Implicit) and existing multi-task learning-based tracker, i.e., JDE [10]

Tracker	Publication	Year	MOT16				MOT17			
			MOTA↑	MOTP↑	IDF1↑	IDs↓	MOTA↑	MOTP↑	IDF1↑	IDs↓
DeepSORT [53]	ICIP	2017	61.4	79.1	62.2	**781**	60.3	79.1	61.2	**2442**
Tracktor+CTdet [34]	ICCV	2019	-	-	-	-	54.4	78.1	56.1	2574
JDE [10]	ICCV	2019	64.4	55.8	1881	-	-	-		
CenterNet [27]	ECCV	2020	-	-	-	-	**67.8**	-	64.7	3039
Chained-Tracker [12]	ECCV	2020	**67.6**	78.4	57.2	1897	66.6	78.2	57.4	5529
Ours(Implicit)	IntelliSys	2021	64.6	78.8	65.9	1234	63.2	78.7	64.8	3357
Ours(Explicit)	IntelliSys	2021	64.9	**79.7**	**66.4**	1489	63.7	**79.1**	**66.0**	3696

Table A.2: Comparing the proposed systems against SoTA two-stage and one-stage trackers that have reported their results on MOT16 and MOT17.

4.2 Implementation Details

The training of the network on a huge dataset is a arduous task and requires a lot of computational power and time. To improve the convergence speed, weights of the proposed models are initialized using a network pre-trained on the COCO dataset. Our network with implicit attention is trained for 60 epochs with a batch size of 16 on 2 × Nvidia 1080 Ti GPUs and took ∼60 hrs to complete. However, our network with explicit attention is trained for 100 epochs with a batch size of 8 on 2 × Nvidia 1080 Ti GPUs and took ∼120 hrs. The collected training dataset is augmented by applying rotation, scaling and color jittering randomly to the input images similar to JDE [10].

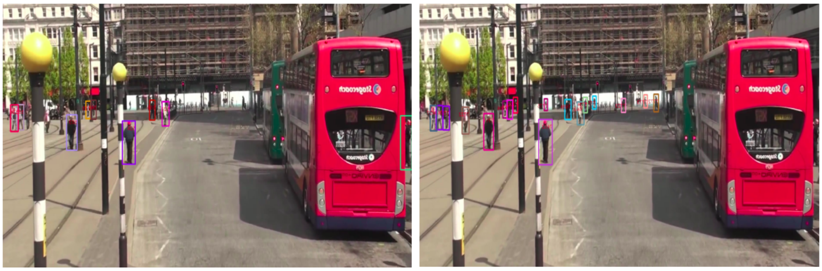


Fig. A.6: left: results obtained from implicit attention-based tracker, **right**: results obtained via our explicit attention-based tracker. Small objects at the back are detected accurately using explicit attention (right), which are missing in left image.

4.3 Experiments

Only a few published approaches, i.e., JDE [10], and Track-RCNN [22] are built on multi-task learning-based one-stage tracking. JDE [10] optimizes two tasks simultaneously, i.e., detection and re-id. However, Track-RCNN [22] jointly optimises detection, re-id, and segmentation task. The inclusion of extra task requires different training datasets containing annotations of segmentation in addition to detection, and re-id, which makes it incomparable with our tracker. We are comparing our method with trackers using private detections, e.g., JDE [10].

Table A.1 compares the existing one-stage tracker's accuracy with our systems using both explicit and implicit attention on MOT15 training dataset as also done in JDE [10]. It can be seen from the table, that our attention-based tracker outperforms the existing one-stage tracker, JDE [10] by a large margin both in terms of MOTA (6.3%) and IDF1 (9.2%).

Table A.2 shows a comparison between the proposed approaches and other SoTA tracking methods reported on MOT challenge [50]. It can be observed from results that our tracker with explicit attention is performing the best in terms of MOTP and IDF1 scores on both MOT16 and MOT17 testing datasets. A high value of MOTP indicates precise localization of target object, while improvement in IDF1 score indicates better prediction of identities compared to Ground Truth (GT). Both ID preservation and precise localization are critical factors for designing a surveillance system as also stated by [46], which enables the proposed attention-gated tracker for security applications. On the other hand, the improvement in overall object detection in this framework increases the size of the affinity matrix, which in turn increases the ID switches (IDs) especially for the crowded sequences.

It can be observed from the quantitative results in Tables A.1 and A.2 that the tracking accuracy is improved by a large margin for the MOT15 validation dataset in comparison to MOT16 and MOT17 test data. The reason is that MOT15 validation sequences are less crowded in comparison to MOT16 and MOT17 test sequences. As discussed before, the proposed system improves the detection of small and partly occluded objects, which increases both the number of objects and the computational complexity of the association matrix. An increase in the size of the affinity matrix leads to multiple ambiguities and hence makes it difficult to optimize for the minimum cost. As a result, the MOTA of our proposed systems is comparatively lower on MOT16 and MOT17 test datasets than the MOT15 validation data containing fewer tracking objects.

4.4 Qualitative Results

Figure A.5 and A.6 show the results of the proposed system using both explicit and implicit attentions. It can be clearly seen from the results in Figure A.6 that the attention modules improve the detection even for small or partially visible objects. Furthermore, results in Figure A.5 illustrate that the proposed attention-gated trackers are performing better in comparison to the existing one-stage tracker JDE [10] (also based on multi-task learning) for the objects approaching towards the boundaries of the frame. In the existing one-stage trackers like JDE [10], inconsistent detections for objects decreases the system's IDF1 score. The proposed systems improve object detection via incorporating attention-gates in the backbone network. However, it also increases the size of the affinity matrix, which further increases identity switches (IDs). One of the main benefits of using attention in the proposed system is that it also improves the embedding vector responsible for assigning an ID to a tracking object, which consequently improves the overall IDF1 of the proposed tracker.

5 Ablation Studies

This section contains the extended experiments required for adapting the base HRNet [13] to the proposed attention-gated architecture. The main experiments require a massive amount of computational time. In order to save computation time, all experiments in this section are trained on a small dataset, i.e., MOT17. The training on a small dataset provides a quick overview of different configurations setting required for explicit and implicit attention-based trackers. This section discusses the details about adding the attention-gates, feature fusion from different abstract levels, and novel modification of network architecture using explicit attention.

5.1 Attention-gates

The feature-maps at the highest resolution in HRNet [13] goes through a series of convolutional operations. As motivated by DenseNets [54], features from every stage are fused with each subsequent stage's features. This fusion can be a simple addition or attention-gated where low-level features are weighted based on high-level features. This helps in extracting only relevant information from the earlier stages. Experimental results in Table A.3 compare the system's accuracy using HRNet [13], dense feature fusion without attention-gates, and attention-gated dense feature fusion. It can be observed from the results that dense feature fusion decreases the accuracy rather than showing any improvement. The main reason for this decrease in accuracy is that the combination of lowest and highest level features decreases the quality of features and hence the overall accuracy. The accuracy is improved by adding attention-gates to the dense-feature fusion but it is still lower than the HRNet [13]. The next section contains experiments to overcome this problem.

Backbone	MOTA↑	IDF1↑
HRNet	**79.2**	**72.9**
Ours(Implicit) + dense-connections + without attention-gates	76.5	72.8
Ours(Implicit) + dense-connections + with attention-gates	**79.2**	72.2

Table A.3: The effect of adding attention-gates during the fusion of features.

5.2 Implicit Attention

The experiments in Table A.3 depict that attention-gated fusion of features improves the accuracy, but dense connections decrease it. Therefore, further experiments are performed by reducing the dense skip connections to consecutive stages as also shown in Figure A.2. In attention-gated dense fusion, features from the current stage are provided to all the subsequent stages at a single resolution. For example, feature maps from stage-1 are provided to stage-2, stage-3, and stage-4 at the highest resolution. However, the attention-gated fusion of features across the consecutive stages only combines the feature from current to the next subsequent stage, e.g., the connection between stage-1 and stage-2. It can be observed from the results obtained in Table A.4 that attention-gated fusion of features across the consecutive stages improves the overall tracking accuracy.

Backbone	MOTA↑	IDF1↑
HRNet	79.2	72.9
Ours(Implicit) + dense-connections	79.2	72.2
Ours(Implicit) + consecutive-stages	**80.7**	**73.9**

Table A.4: Comparing the tracking accuracy by adding attention-gated dense connections and consecutive-connection across the different stages.

5.3 Explicit Attention

The proposed explicit attention is guided by additional supervision by minimizing the auxiliary loss at an intermediate stage as shown in Figure A.3. This auxiliary loss minimizes the deviation between predicted and GT heatmaps. The heatmap is predicted using attention-gated features at different stages of the network. The results in Table A.5 which shows adding an auxiliary loss after stage-1 has a negligible effect on overall tracking accuracy. As we shift the auxiliary loss towards the later stages of the network, the overall tracking accuracy starts increasing. This implies that the initial features of HRNet [13] does not contain any information about the global context and hence are incapable of making any predictions. The feature-maps get more relevant when the global context increases by an additional level after every stage. Additionally, this auxiliary loss is calculated only for the feature-maps generated from the high resolution because the lower resolutions are added at the later stages of the network. Furthermore, we would like to deal with inconsistencies in the detection of smaller objects which are detected at higher resolutions.

Backbone	MOTA↑	IDF1↑
HRNet	79.2	72.9
Ours(Explicit) + Stage 1	79.2	73.5
Ours(Explicit) + Stage 2	79.6	73.7
Ours(Explicit) + Stage 3	**81.8**	**74.8**

Table A.5: Adding supervised attention and auxiliary losses after stage 1, 2 and 3 at highest resolution in HRNet [13].

6 Conclusion

In this paper, we proposed two one-stage trackers based on implicit and explicit attention mechanisms for multi-object tracking. The one based on implicit attention utilizes a novel fusion of feature maps for combining information extracted from different abstract levels, while the other based on explicit attention utilizes an auxiliary heatmap function that provides additional supervision for the attention mechanism. The latter tracker outperforms the former one, and both outperform state-of-the-art tracking systems in terms of MOTP and IDF1 scores when evaluated on public benchmark datasets. We observed that our proposed attention-based architectures improve the tracking accuracy for less crowded scenes such as MOT15 sequences. For crowded sequences such as MOT16 and MOT17, the proposed attention-gated one-stage tracker improved the feature propagation and hence the object detection and identity-based measures such as IDF1 scores. However, to improve MOTA on MOT16 and MOT17, as a future work, we will work on developing an association block to reduce the identity switches.

Acknowledgements

This work was supported by the Milestone Research Programme at Aalborg University (MRPA).

References

[1] Y. Huang, I. Liao, C. Chen, T. İk, and W. Peng, “Tracknet: A deep learning network for tracking high-speed and tiny objects in sports applications*,” in *16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–8.

[2] A. Bera, S. Kim, and D. Manocha, “Realtime anomaly detection using trajectory-level crowd behavior learning,” in *CVPRW*, 2016, pp. 1289–1296.

[3] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, “Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing,” in *American Control Conference*, 2007, pp. 2296–2301.

[4] J. M. Fitts, “Precision correlation tracking via optimal weighting functions,” in *1979 18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, vol. 2, 1979, pp. 280–283.

[5] K. Fukunaga and L. Hostetler, “The estimation of the gradient of a density function, with applications in pattern recognition,” *IEEE Transactions on Information Theory*, vol. 21, no. 1, pp. 32–40, 1975.

[6] L. Bertinetto, J. Valmadre, F. J. Henriques, A. Vedaldi, and H. S. P. Torr, “Fully-convolutional siamese networks for object tracking,” *ECCV Workshops*, 2016.

[7] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, “Spatially supervised recurrent convolutional neural networks for visual object tracking,” *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2017.

[8] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. H. Lau, and M. Yang, “Vital: Visual tracking via adversarial learning,” in *CVPR*, 2018, pp. 8990–8999.

[9] H. W. Kuhn and B. Yaw, “The hungarian method for the assignment problem,” *Naval Res. Logist. Quart*, pp. 83–97, 1955.

[10] Z. Wang, L. Zheng, Y. Liu, and S. Wang, “Towards real-time multi-object tracking,” in *ECCV*, 2020.

[11] A. Kendall, Y. Gal, and R. Cipolla, “Multi-task learning using uncertainty to weigh losses for scene geometry and semantics,” in *CVPR*, 2018.

[12] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, “Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking,” in *Proceedings of the European Conference on Computer Vision*, 2020.

[13] 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,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[14] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, and D. Rueckert, “Attention gated networks: Learning to leverage salient regions in medical images,” *arXiv*, 2019.

[15] S. Jetley, N. A. Lord, N. Lee, and P. Torr, “Learn to pay attention,” in *International Conference on Learning Representations*, 2018.

[16] H. Guo, K. Zheng, X. Fan, H. Yu, and S. Wang, “Visual attention consistency under image transforms for multi-label image classification,” in *CVPR*, 2019.

[17] Z. Huang, S. Liang, M. Liang, and H. Yang, “Dianet: Dense-and-implicit attention network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 4206–4214.

[18] C. Liu, J. Mao, F. Sha, and L. A. Yuille, “Attention correctness in neural image captioning,” *AAAI*, 2017.

[19] C. He and H. Hu, “Image captioning with visual-semantic double attention,” *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 15, no. 1, 2019.

[20] K. Li, Z. Wu, K.-C. Peng, J. Ernst, and Y. Fu, “Tell me where to look: Guided attention inference network,” in *CVPR*, 2018, pp. 9215–9223.

[21] J. Gao, T. Zhang, and C. Xu, “Graph convolutional tracking,” in *CVPR*, 2019, pp. 4644–4654.

[22] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, “Mots: Multi-object tracking and segmentation,” in *CVPR*, 2019, pp. 7934–7943.

[23] S. Jin, W. Liu, W. Ouyang, and C. Qian, “Multi-person articulated tracking with spatial and temporal embeddings,” in *CVPR*, 2019.

[24] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, “Poi: Multiple object tracking with high performance detection and appearance feature,” *ECCV Workshops*, 2016.

[25] P. Chu and H. Ling, “Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking,” in *ICCV*, 2019, pp. 6171–6180.

[26] F. Yu, W. Li, Q. Li, Y. Liu, X. Shi, and J. Yan, “Poi: Multiple object tracking with high performance detection and appearance feature,” in *ECCV Workshops*, 2016.

[27] X. Zhou, V. Koltun, and P. Krähenbühl, “Tracking objects as points,” *ECCV*, 2020.

[28] 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.

[29] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *arXiv*, 2015.

[30] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *International Conference on Neural Information Processing Systems*, 2015, p. 91–99.

[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, “SSD: single shot multibox detector,” in *ECCV*, 2016, pp. 21–37.

[32] A. Jadhav, P. Mukherjee, V. Kaushik, and B. Lall, “Aerial multi-object tracking by detection using deep association networks,” in *2020 National Conference on Communications (NCC)*, 2020, pp. 1–6.

[33] K. Kang, H. Li, T. Xiao, W. Ouyang, J. Yan, X. Liu, and X. Wang, “Object detection in videos with tubelet proposal networks,” *CVPR*, 2017.

[34] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, “Tracking without bells and whistles,” in *ICCV*, 2019.

[35] A. Graves, “Generating sequences with recurrent neural networks.” *arXiv*, 2013.

[36] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” *arXiv*, 2014.

[37] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in *CVPR*, 2019.

[38] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *CVPR*, 2018.

[39] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, “Residual attention network for image classification,” in *CVPR*, 2017.

[40] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “Cbam: Convolutional block attention module,” in *ECCV*, 2018.

[41] O. Oktay, J. Schlemper, L. L. Folgoc, M. J. Lee, M. Heinrich, K. Misawa, K. Mori, S. G. McDonagh, N. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, “Attention u-net: Learning where to look for the pancreas,” *ArXiv*, 2018.

[42] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “A simple baseline for multi-object tracking,” *arXiv*, 2020.

[43] K. Sun, Z. Geng, D. Meng, B. Xiao, D. Liu, Z. Zhang, and J. Wang, “Bottom-up human pose estimation by ranking heatmap-guided adaptive keypoint estimates,” *arXiv*, 2020.

[44] G. Welch and G. Bishop, “An introduction to the kalman filter,” 1995.

[45] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: The CLEAR MOT metrics,” *EURASIP Journal on Image and Video Processing*, pp. 1–10, 2008.

[46] E. Ristani, F. Solera, R. S. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in *ECCV Workshops*, 2016, pp. 17–35.

[47] A. Ess, B. Leibe, K. Schindler, , and L. van Gool, “A mobile vision system for robust multi-person tracking,” in *CVPR*, 2008.

[48] S. Zhang, R. Benenson, and B. Schiele, “Citypersons: A diverse dataset for pedestrian detection,” in *CVPR*, 2017.

[49] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” in *CVPR*, 2009.

[50] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, “Mot16: A benchmark for multi-object tracking,” *arXiv*, 2016.

[51] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang, “Joint detection and identification feature learning for person search,” in *CVPR*, 2017.

[52] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang, and Q. Tian, “Person re-identification in the wild,” in *CVPR*, 2017, pp. 1367–1376.

[53] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[54] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *CVPR*, 2017, pp. 2261–2269.

Paper B

ThermalSynth: A Novel Approach for Generating Synthetic Thermal Human Scenarios

Neelu Madan, Mia Sandra Nicole Siemon, Magnus Kaufmann Gjerde,
Bastian Starup Petersson, Arijus Grotuzas, Malthe Aaholm Esbensen,
Ivan Adriyanov Nikolov, Mark Philip Philipsen, Kamal Nasrollahi,
Thomas B. Moeslund

The paper has been published in the
Winter Conference on Applications of Computer Vision Workshops (WACVW), pp.
130-139, 2021.

The layout has been revised.

Abstract

In this paper, we propose ThermalSynth, a novel approach for creating synthetic thermal images by mixing 3D characters generated using the Unity game engine with real thermal backgrounds. We use a shader based on the Stefan-Boltzmann law [1] to approximate the appearance in the thermal domain of the synthetic characters. Additionally, we provide a post-processing pipeline to better blend the high-fidelity synthetic data with the lower-resolution real thermal surveillance one. The proposed approach is used to create a dataset for people falling into water near a harbor front. Diverse scenarios of such falls are generated with an ample amount of data to enable the use of deep learning algorithms. To demonstrate the effectiveness of the generated data, we train two standard deep neural networks (AlexNet and ResNet-18) on our synthetic thermal dataset using a supervised learning approach. We test our system on small datasets containing real video footage of actual falls. We observe that training these simple classification networks yields an accuracy of 98.70% *at a sensitivity of* 100% *on the real-world voluntary fall dataset. The code for ThermalSynth and the dataset is publically available at https://github.com/NeeluMadan/Thermal-Synth.*

1 Introduction

Video surveillance systems mostly employ stationary RGB cameras as they are cost-effective whilst yielding good discrimination among objects. They are, however, not immune to various quality-decreasing conditions such as occlusions, changing weather, and low illumination. Thermal cameras, on the other hand, measure the difference in heat signatures returning high-contrast images. Consequently, they are a reliable choice under diverse weather conditions while preserving privacy [2] at the same time, which also helps comply with the General Data Protection Regulation (GDPR) [3, 4]. As a result, intelligent video surveillance systems have started using a combination of RGB and thermal cameras, recently.

In the RGB domain, there exist multiple instances of synthetic datasets [5–9] which were generated using a virtual environment. However, only a very limited amount of research focuses on generating synthetic datasets in the thermal domain. In this paper, we propose a pipeline to generate synthetic thermal datasets. Generating such data gives the possibility to address different scenarios with very few instances in real life that are hard to replicate through physical testing setups. One such scenario is the depiction of humans falling into bodies of water in an outdoor environment. As these instances occur rarely and often under life-threatening conditions, it is very difficult to obtain data of such scenarios. Only a few datasets for this problem currently exist containing voluntary falls [10] and the use of dummies [11]. The foremost drawback of generating such data via intentional jumps [10] is that it might get intermittently dangerous due to unpredictable circumstances. The use of a dummy [11] on the other hand, requires a time-consuming preparation process and also has only a limited degree of freedom to add variations. As a consequence, it is difficult to obtain

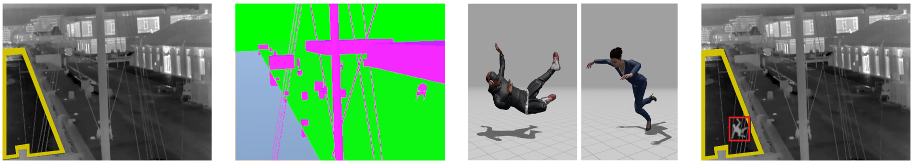

(a) Background (b) Unity Scene (c) Mixamo Characters (d) Fall Scene

Fig. B.1: ***ThermalSynth*** **- Synthetic data generation proposal** B.1a) An example background image from the Long-term Thermal Drift (LTD) dataset [12], B.1b) the same scene synthesized in Unity, B.1c) example fall animations of Mixamo characters, and B.1d) a synthetically generated falling person merged with B.1a, with a yellow enclosure highlighting the Region of Interest (RoI).

a large and varied dataset using either method, rendering deep neural networks and/or machine learning models inapplicable to this problem domain. We therefore propose an approach to generate a synthetic thermal datasets, and apply it to the concrete use-case of human fall detection at harbor fronts. It is important to mention, that given the modular nature of our proposed data generation approach, it could be easily modified to create synthetic thermal datasets for almost any application domain, with the only requirement of providing initial real-life images of the environment that can be used as background.

For demonstration purposes, we trained two classic Convolutional Neural Networks, AlexNet [13] and ResNet-18 [14], using supervised learning on our proposed synthetic fall dataset and tested the model on real [10] and semi-real (dummy) datasets [11]. The best model results in 98.70% accuracy and a sensitivity rate of 100% on real data constituting of intentional falls, indicating that our synthetic dataset contains a good approximation of the distribution of real-world human fall scenarios. The contribution of this work to video surveillance in the thermal domain is two-fold:

1. A synthetic data generation pipeline in which uniquely generated foreground objects are combined with real background footage

2. The application of our proposed pipeline to generate a synthetic fall dataset for people falling into water

The rest of the paper is structured as follows: The next section provides an overview of existing research in synthetic datasets and the fall detection domain. Section 3 describes the data generation process and the standard classification models used in this research. Our experimental setup is mentioned in Section 4, which is then followed by results and discussion in Section 5. We finally conclude our research with its possible future directions in Section 6.

2 Related Work

Synthetic Datasets in Thermal Domain. The enforcement of the new GDPR [3, 4] law by the European Union has turned the acquisition of large-scale personal visual

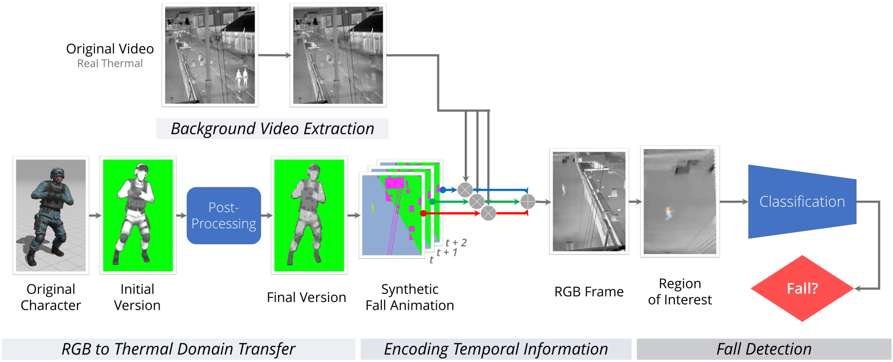


Fig. B.2: Synthetic Data Generation and Fall Detection Pipeline: Our synthetic data is limited to three frames at $t, t+1, t+2$. After merging each of them individually with the extracted background frames, they are stacked together in order to encode temporal information.

data into a challenging task. Under these circumstances, and given the fact that deep learning networks are very data-hungry, we have experienced a paradigm shift towards the generation of synthetic datasets for sensitive data. In the thermal spectrum domain, there exist two methods for generating synthetic data: (1) Mapping straight from the RGB domain, and (2) using virtual environment engines.

In the domain of deep learning, the use of approach (1) is commonly achieved by means of Generative Adversarial Networks (GANs). This takes either place in a supervised (paired data) or an unsupervised (unpaired data) setting [15–19]. Research [15, 17] shows, however, that the usage of supervised GANs [17, 19] delivers better results than its unsupervised counterpart [18] because of the presence of RGB-thermal image pairs. Since we only have thermal video footage at our disposal for the purpose of this project though, the usage of supervised GANs is not applicable.

With respect to approach (2), only a few instances generating thermal datasets synthetically using virtual environments [9, 20, 21] exist to date, even though there are numerous such environments for generating synthetic data in the visual spectrum: CARLA used for Advance Driver Assistance System (ADAS) [22], VIVID [23] for indoor navigation, Gazebo [24] for simulating multi-robot, and Habitat 2.0 [25] for home assistants. Apart from that, other research [9, 20, 21] uses game engines such as Unity [26] and Unreal [27] to generate photo-realistic synthetic data. Pramerdorfer *et al.* [20], for instance, generate synthetic depth and thermal images showing human behavior captured in indoor environments using Blender [28] while Blythman *et al.* [9] generate synthetic thermal human heads placed in cars using Zephyr [29]. Bongini *et al.* [21] use Unity [26] to generate synthetic thermal videos by combining 3D foreground objects with the real background images in autonomous driving scenarios. Following this trend of synthetic data generation in the thermal spectrum domain, we consequently propose to generate a synthetic thermal dataset for human fall detection using Unity [26]. Similarly to the work in [21], the proposed approach generates our dataset images by blending synthetic foreground objects with real background scenes. These are obtained from the Long-term Thermal Drift (LTD) dataset [12].

Fall Detection. Fall incidents in outdoor scenarios, like people falling into water [10, 11, 30] or on the street [31, 32], have only received little attention in recent years. There exists qualitative research [31, 32] addressing cases of falls in outdoor scenarios such as sidewalks, streets, and garages, but it still remains difficult to capture such moments using surveillance cameras. One reason for this is that there is an insufficient amount of samples available, especially when it comes to people falling into water. One of the first works to address the problem was done by Bonderup *et al.* [10]. Here, a pipeline consisting of person detection, person tracking, fall prediction, and fall detection is proposed in the thermal domain. Additionally, Bonderup *et al.* [10] generated a thermal fall dataset asking people to perform intentional jumps into water. A few years later, Nikolov *et al.* [11] proposed a semi-real dataset for human fall detection simulating fall scenarios in the thermal domain using a dummy. The authors make use of calculated optical flow maps around a specific area of interest in order to detect falls.

In this research, we address the human fall detection problem using a supervised classification approach. The related works described so far suffer from significant limitations as the datasets are captured in controlled conditions and hence contain only very limited variations. Our intention is to fill this gap and present a synthetic dataset that can serve as an extensive source of diverse human fall scenarios.

3 *ThermalSynth*: Proposed Method

The main objective of this research is to present an approach to create a pipeline for synthetically generating thermal datasets using a 3D environment. We also demonstrate its application for creating synthetic human falls into water regions at harbor fronts. This section contains the building blocks of *ThermalSynth*, our thermal synthetic data generation process followed by its application for human fall detection. The entire pipeline for our proposed method is shown in Figure B.2.

Images are generated by merging real thermal backgrounds with synthetic people generated in Unity [26]. The real videos are obtained from the LTD dataset [12], which is very diverse in terms of different weather conditions as it encapsulates video data for 8 months (January-August) from a single camera view. Such single-scene recordings are most common in video surveillance setups. The main elements of our synthetic data generation pipeline are as follows: (1) Background Extraction, (2) Foreground Generation, (3) Thermal Shader and (4) Post-Processing. Each of these steps is explained in detail in the following subsections.

3.1 Background Extraction

All background scenes are extracted from the LTD dataset [12]. It contains 298 hours of single-scene videos, with a resolution of 384 × 288, and captures a single camera view. Each video is 2 minutes long, and all are uniformly spaced out throughout 24 hours of the day. In order to create the backgrounds, a temporal median filter is applied to the dataset. This is done as a two-step process: At first, the videos are coarsely sampled at 1 frame per second (FPS) in order to manage the computational complexity. Secondly, the median value for each matching pixel across all frames for each 2-minute video is computed. The temporal median filter is solely applied to the harbor area, in order to retain other movements caused by the water and other moving objects close to the camera due to wind like wires, ropes, and masts. This is done by manually creating a mask of the parts of the scene where the filter should be applied. To limit the complexity of our dataset, we uniformly sample 69 hours of video from the LTD dataset to extract the backgrounds. The sequence of those extracted background frames is kept as given in the original video in order to retain the fluency of non-object motions such as clouds, waves, and wires. We stacked three consecutive frames together, which are later blended with the synthetic foreground as also shown in Figure B.2.

3.2 Foreground Generation

For the generation of our synthetic thermal foreground videos the game engine Unity [26] was chosen. Depicting different scenarios of people walking and falling into water at the harbor, these synthetic videos are later merged with the created background instances. Generating synthetic videos is also a two-part process. First, the 3D models of people are selected together with a number of animations for walking, running, jumping, falling, etc. Mixamo [33] is used to select these 3D models and animations, as it is a free-to-use library of human-looking characters, together with motion-captured animations. Examples of some of the character models used are shown in Figure B.1c. We choose a total set of $79,998$ unique foreground video sequences depicting jumps and falls, each comprising three consecutive frames.

As the next step, the parameters of the chosen thermal camera for recording the LTD dataset (see [34] for details) are transferred to the Unity camera. To do so, the Universal Render Pipeline in Unity is used together with the physical camera settings. The parameters are given in Table B.1. A synthetic scene is then modeled with primitive objects in Unity in places where real objects can obscure the view of the camera of people walking on the street. Real-world objects deemed necessary to be modeled are selected heuristically after observing videos from the LTD dataset. The Unity camera's position and orientation are then set to best match the position of the real-world one. The resulting synthetic scene in Unity can be seen in Figure B.1b, where the modeled obscuring objects are shown in pink, the background from the real images in green, and the waterfront in gray, together with a synthetic person falling. The real

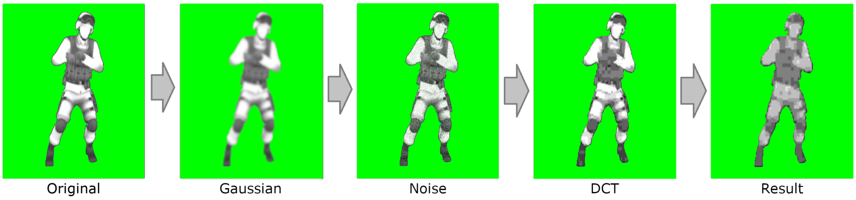


Fig. B.3: Post-processing Pipeline: Starting from left the original image generated through applying a thermal shader on a character taken from Mixamo [33], Gaussian filter with kernel size 3x3, random noise applied, and finally DCT compression artifacts added based on Kane *et. al.* [1]. The final result is the sum of all single instances.

background is visualized on a rendered texture behind the synthetic scene. We then use the Perception package [35] provided by Unity [26], to generate a large number of combinations of 3D meshes, animations, and backgrounds. These masks of rendered people are used in the post-processing step to better blend the synthetic foreground with the background. The next step is to transform the generated synthetic pedestrian footage from RGB to thermal domain using a custom shader. An overview of the steps for creating the shader is given in the next section.

3.3 Thermal Shader

Once the synthetic pedestrians (foreground) are generated, together with their segmentation masks, their RGB representation needs to be transformed into a thermal one. The thermal shader used in our approach is inspired by Kane *et al.* [1]. It uses the Stefan-Boltzmann law to compute the black body radiation (j^*) of an object given by Equation B.1, where T is the absolute temperature of the object in Kelvins, ϵ_m represents the thermal emissivity of the material, and σ is the Stefan-Boltzmann constant equalling to $5.6704 \times 10^{-8} \frac{W}{m^2K^4}$.

$$j^* = \epsilon_m \, \sigma \, T^4 \tag{B.1}$$

To translate this in the context of a shader we first find the emissivity values of some of the materials that would be part of the generated pedestrians. In our case, we simplify this to human skin and clothes. Values for both of these are given in the article

Thermal Camera				
Zoom	Resolution	Frame Rate	Lens	FOV
Fixed	384 × 288	25/30 FPS	25mm	21.7°

Emissivity Values				
Human Skin	Cotton	Asphalt	Water	Snow
0.95	0.95	0.95	0.93	0.90

Table B.1: Parameters of thermal camera used in the LTD dataset [12], Emissivity coefficients of materials found in our scenes, taken from [1]

written by Kane *et al.* [1]. We have listed these together with the emissivity values of other materials for comparison in Table B.1. Next, for the absolute temperature in Kelvin, we select an average value of 300.5 for simplifying the generation process.

Once we have these initial values, we follow the approach described by Kane *et al.*. The albedo texture color of each pixel is transformed to luminance (L) using Equation B.2, where R, G, and B, are the red, green, and blue color channels, respectively.

$$L = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \tag{B.2}$$

As presented in the article [1], the calculated luminance is then used to approximate color emissivity (ϵ_c) using Equation B.3, by using the average color emissivity of a white color surface of 0.84 and the percent difference between white and black object emissivity of 0.15.

$$\epsilon_c = (1 - L) \cdot 0.15 + 0.84 \tag{B.3}$$

The material and color emissivities are then blended using a blend factor of 0.31. The final blended value is further used in the Stefan-Boltzmann equation shown in B.1. Finally, the calculated thermal radiation value for each pixel is mapped to an intensity range between $[0, 1]$ so it can be displayed by Unity. A gain (G) and level (L) control are made available for the final pixel value p using Equation B.4, so that manual adjustment can be possible. For the purpose of this paper, these values were manually set to $G = 0.05$ and $L = 20$ as these provided the blending with the extracted backgrounds (this effect is visualized in Figure B.5).

$$p = (j^* \cdot G) + L \tag{B.4}$$

Once the foreground pedestrian footage is transformed into thermal, the next step is to blend it with the extracted real backgrounds. To do this, a number of post-processing steps are implemented which are discussed in the next section.

3.4 Post-Processing

In practice, thermal camera sensors are susceptible to capturing noise from the environment [1], which together with compression artifacts from storing videos may

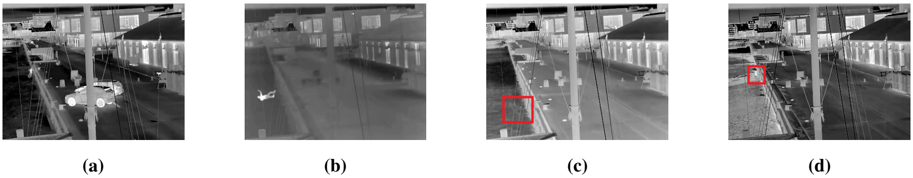

(a) (b) (c) (d)

Fig. B.4: Results of synthetic thermal frame generation pipeline: Successful frames are shown in B.4a and B.4b where foreground objects, i.e., humans can be distinguished from background, and unsuccessful ones in B.4c and B.4d where foreground can't be discriminated from background.

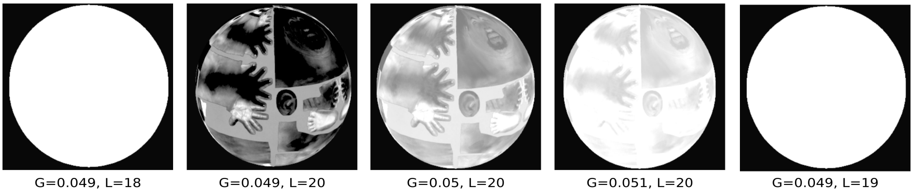


Fig. B.5: Illustrating the effect of different Gain (G) and Level (L) ranges on the appearance of the foreground. G=0.05 and L=20 generate the most realistic thermal appearance.

degrade the visual quality of captured footage. Bhatia *et al.* [36] propose a post-processing stack of image effects for simulating infra-red sensors and their specific characteristics. We choose three prominent effects based on Kane *et. al.* [1] from those - blurring, random noise, and compression artifacts. Together with the already implemented part of the thermal shader with gain and level processing, these effects help blending the real and synthetic parts of the image into one coherent picture.

1. **Gaussian blur:** Rendered objects in Mixamo [33] contain sharp or jagged edges in comparison to real objects. Blurring artifacts are therefore introduced to the area around the synthetic humans using a (3 × 3) Gaussian kernel.

2. **Random noise:** The degree of sensitivity of image sensors used to capture real-life footage by means of thermal cameras often introduces random noise. Mitigation of this effect is achieved through the application of random noise to our rendered figures by generating uniformly distributed random numbers, and through linear interpolation between this value and the rendered foreground one.

3. **Compression artifacts:** Mosquito Noise and Block Artifacts appear to be the most common side-effects caused by flawed compression algorithms that are implemented in thermal cameras which make use of block-based Discrete Cosine Transform (DCT) [37]. In order to account for this imperfection, additional encoding and decoding of the foreground character become necessary. Hence, the image is firstly converted into JPEG format, which performs DCT compression by default, setting the quality value to 5 (on a scale from 0 to 100), and secondly it is decoded to retrieve its compressed form.

4. **Compositing:** To blend the post-processed synthetic person into the real background image, screen compositing mode is used. Being a compositing technique that preserves the edges of the foreground mask, for images that come in 8-bit integer precision the composited image can be calculated using Equation B.5:

$$C = 255 - \frac{(255 - F) \times (255 - B)}{255}, \tag{B.5}$$

with C equalling the composited image, B to the real background, and F to the synthetic foreground. Example final results from the synthetic thermal image generation including post-processing that is visualized in Figure B.3 can be seen in Figure B.4. Success and failure are determined based on how well the synthetic foreground elements blend visually into the real background scenery. In comparison, decisive attributes encompass the level of visibility of the foreground compared to the background, and differentiation between different body parts, such as extremities, torso, and head, for example.

3.5 ThermalSynth for Human Fall Detection

We use the pipeline explained earlier in this section to generate a synthetic thermal dataset of humans falling into water. This dataset is kept very simplistic by restricting it to a limited number of human fall animations, as it is primarily serving demonstration purposes. We further use this dataset to train two classic Convolutional Neural Networks, AlexNet [13] and ResNet-18 [14], using supervised learning in order to perform human fall detection. The fall detection problem is modeled here as a binary classification one with our two classes being defined as *fall* and *no fall*, respectively. As mentioned earlier in this paper, *ThermalSynth* is not limited to this particular application area. Thanks to the generic design, it is applicable to a wide range of surveillance scenarios that take place in the thermal domain.
Besides privacy, another major advantage of using synthetic datasets for training machine learning models is that annotations can be generated automatically as part of the data creation process. These automatic labels are highly accurate in comparison to manually annotated ones. The absence of such noise during the training process of machine learning models results in more robust prediction performances. Based on the achieved results described in the upcoming section it can be observed that models trained on synthetic data perform almost perfectly when tested on real-world data.

4 Experiments

4.1 Datasets

We evaluate the performance of our models on two datasets with real thermal surveillance footage: *Intentional Fall Dataset* (real) [10] and *Dummy Dataset* (semi-real) [11]. Both of these datasets contain only a very limited number of samples and thus would result in severe overfitting when used to train a neural network. We therefore use our proposed synthetic dataset for training the models instead, and use the other two datasets for test purposes only.

Intentional Fall Dataset The Intentional Fall dataset was collected by Bonderup *et al.* [10]. It was recorded in the thermal domain and visualizes scenarios of a variety of jumps into water performed by volunteers. Out of the manually annotated thermal video footage (captured during Spring 2016) a subset was chosen that depicts the same harbor scene as the LTD dataset [12]. On concatenation of three consecutive frames into a single batch as an RGB image, the test set ends up consisting of 77 samples in total, out of which 18 are denoted as *fall*, and 59 as *no fall*.

Dummy Dataset The Dummy dataset, interchangeably also called mannequin or rubber doll dataset by its authors [11], was introduced in order to show that an air-filled rubber doll presents a sufficient representation of humans when generating thermal video footage that targets the detection of human falls into water. The authors of [11] generated a thermal video dataset (captured during the months of September - October 2021) that depicts artificially arranged emergencies at a harbor front. For the sake of this work, the videos are also parsed in a way that allows for the compression of three consecutive frames into a single batch in form of an RGB image. This leads to a Dummy test set which consists of $1,626$ frames out of which 580 were categorized as *fall*, and $1,046$ as *no fall*.

4.2 Evaluation Metrics

All our models are evaluated in terms of sensitivity, specificity, and accuracy. *Accuracy* describes correctly classified *fall*s and *no fall*s over all cases. *Sensitivity* describes correctly detected *fall*s over all *fall* cases. *Specificity* on the other hand concerns correctly classified *no fall*s over all *no fall*. For the purpose of solving fall detection tasks, those systems with high sensitivity are preferable as it is crucial to detect as many *fall* cases as possible even at the expense of falsely classifying few *no fall* cases. Not achieving this, i.e., missing *fall*s, may possibly result in a person drowning.

4.3 Implementation Details

Since fall detection in water regions can be considered as a special scenario of binary classification, we employ two standard classification networks, i.e., AlexNet [13] and ResNet-18 [14], which are trained solely on the proposed synthetic thermal data. Due to the simplicity of the problem, we refrain from proceeding with more complex architectures at this leads to overfitting, and a significant decrease of the system's performance.

Training. Before launching the training of the networks, the generated synthetic images are pre-processed by cropping only the water area. Since this results in a trapezoidal image, it is further warped using *warpPerspective* function from OpenCV to convert it into a rectangular shape. Afterwards, three consecutive frames are concatenated to generate a single tensor of size $185 \times 115 \times 3$. An equal number of *fall* and *no fall* images (69,000) are used for training the baseline models. For the validation of our classification networks, 34,596 images (with 17,298 *fall* and 17,298 *no*

Network	Accuracy (↑)		
	Synthetic Dataset	Intentional Fall Dataset	Dummy Dataset
AlexNet	**99.52**	**98.70**	75.00
ResNet-18	98.75	89.61	75.00

Table B.2: Comparing accuracy in % of the two baseline classification networks. The networks are trained on our proposed Synthetic dataset and tested on a different subset of the Synthetic dataset, together with the full Intentional Fall, and Dummy datasets.

Network	Intentional Fall		Dummy	
	Sens.(↑)	Spec.(↑)	Sens.(↑)	Spec.(↑)
AlexNet	**100.00**	98.00	41.00	**98.00**
ResNet-18	94.00	88.00	**57.00**	88.00

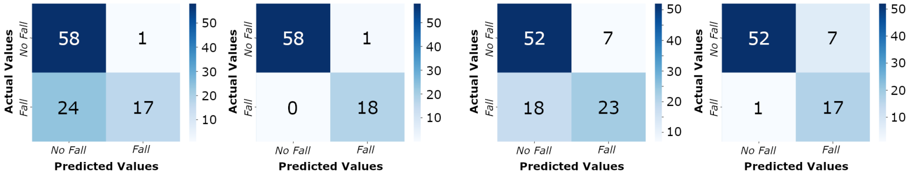
Table B.3: Comparing Sensitivity (Sens.) and Specificity (Spec.) in % of the two baseline classification networks trained on our proposed synthetic dataset and tested on Intentional Fall and Dummy dataset, respectively. Higher values indicate better systems.

fall) are sampled, and the model with the best validation accuracy is saved for further evaluation. The used architectures converged at epoch 20 using a batch size of 32. Stochastic Gradient Descent (SGD) [38] with a learning rate of 10^{-3} is used for optimizing the neural network. PyTorch implementations of our baseline models were chosen and trained using a single NVIDIA RTX 2080 Ti series GPU.

Testing. The performance of our models is evaluated on three test sets coming from three different sources [10, 11] including ours, described in Subsection 4.1. Both datasets for people fall detection, i.e., Intentional Fall [10] and Dummy [11], do not contain any annotations for fall classification. These were created by means of manual frame-level annotations categorizing them as *fall* or *no fall*, respectively.

5 Results and Discussion

The evaluation of our fall detection approach leads to the results recorded in Tables B.2 and B.3. These numbers prove that our synthetically generated data provide a good approximation of the distribution of fall scenarios in the existing datasets and hence



(a) AlexNet (D) **(b)** AlexNet (IF) **(c)** ResNet-18 (D) **(d)** ResNet-18 (IF)

Fig. B.6: Confusion Matrices for AlexNet and ResNet-18 during tests on the Dummy (D) and the Intentional Fall (IF) Datasets.

constitute a justified choice for training deep neural networks. Significant growth in classification accuracy is observed when we use the synthetic thermal dataset for training and testing. We are, however, unable to reach a similar performance when testing on the Dummy dataset, where both network models reached a 75% classification accuracy. The assumption is that the dummy constitutes a very limited representation of a human when falling/thrown into water. This is additionally supported by the results which are given in Table B.3. It can be seen that both networks achieve high specificity on both the Intentional Fall and Dummy datasets. This, however, comes with the caveat that both datasets can be described in a binary way and consist of only frames of either people and dummies falling or not falling into water. The datasets do not contain the third category of objects being in the water without being classified as a person falling into water. In real-world use cases, this category has a very strong possibility of emerging - for example birds landing in the water, boats passing in front of the camera, people throwing objects into water, etc. From this perspective, the high specificity and accuracy of our results should be viewed as *ideal* cases. Automated emergency detection systems need to be robust against false positive detections, as a high number of these can result in diverting resources and drowning out real ones in noise.

Figure B.6 illustrates the confusion matrices determining correct and incorrect classifications in case of fall and non-fall events for both Dummy and Intentional Fall datasets. Looking at these values proves that both our models have great difficulties when having to correctly classify actual *fall* images as *fall* when tested on the Dummy dataset. In other words, out of 41 falls given in this dataset, AlexNet is capable of correctly classifying only 17 of them as *fall* whilst ResNet-18 is performing slightly better with a total of 23 *fall*s. Comparing qualitative classification performance on the Intentional Fall dataset, however, indicates that AlexNet would be the ideal candidate leaving no falls undetected.

Last but not least, we would like to address the choices made with respect to emissivity values during this research. In contrast to the source of values reported in Table B.1, i.e., [1], previous works [39, 40] have shown that these values can be taken from a great range of possibilities: Cotton, wool, and PET, for instance, lay between 0.7 and 0.83, as shown by [39]. In this work, we choose a consistent value of 0.972 to make the proposed synthetic dataset appear visually closest to the thermal domain. Additionally, we apply this emissivity value uniformly to the entire foreground image for simplification purposes.

In addition to the implicit judging of the quality of our dataset by training deep neural networks on it and testing on real-world surveillance footage, we also verify the quality-level of our data implicitly via visual inspection. Some examples of synthetic humans and real humans as foreground objects are shown in Figure B.7. The synthetic humans in Figure B.7 (left) contain the same overall texture from head to toe, whereas real humans shown in Figure B.7 (right) show variations in appearance based on the types of clothes and additional accessories, e.g., bags. We plan to extend this work by applying a part-based thermal shader, where different emissivity values to different

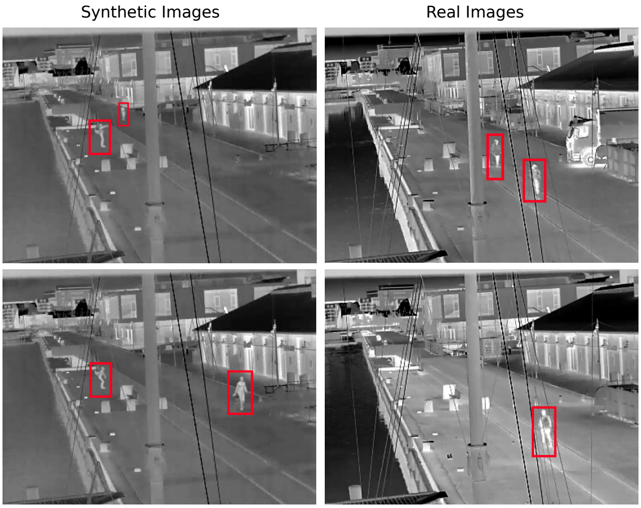


Fig. B.7: Real vs. Synthetic Left: Final synthetic images; Right: Real frames taken from the LTD dataset [12]

.

parts of the foreground are applied. The example of different parts when we consider humans as our foreground objects are head, torso, legs etc.

6 Conclusion and Future Work

In this paper, we introduced *ThermalSynth*, a pipeline for creating synthetic thermal images showcasing one possible application of people falling into the water. For generating the foregrounds we use Unity together with rigged, animated 3D models and a custom thermal shader based on the black body radiation equations along with the Stefan-Boltzmann law. To mimic CCTV camera footage, we implemented a four-stage post-processing pipeline which introduces additional image distortion and finally blends the foreground and background parts. We use this pipeline to create a synthetic thermal dataset of people falling into the water and further train two standard classification models, AlexNet and ResNet-18, to detect fall cases. We test the models on a combination of the synthetic 3D model falls, real-person falls, and simulated falls using a dummy. We show that the standard models achieve very good results in the given context proving the usability and potential of *ThermalSynth* for creating rarely observed emergency scenarios and enriching existing real thermal datasets with synthetic data.

A possible negative societal impact of this dataset is that it reveals different jumping and falling behavior patterns of humans. At the same time, however, this study could save many human lives. We plan to extend this research by introducing additional 3D models to the generation pipeline like vehicles, boats, birds, moving parts of the foreground, etc. in order to possibly use the dataset for multi-class classification tasks and more robust emergency detection in real-life production scenarios.

Acknowledgements

This work was supported by the Milestone Research Programme at Aalborg University (MRPA).

References

[1] F. Kane, “Simulation of night-vision and infrared sensors,” in *Game Engine Gems 2*, E. Lengyel, Ed. A K Peters, 2011, pp. 45–54.

[2] M. Kieu, A. D. Bagdanov, M. Bertini, and A. Del Bimbo, *Domain Adaptation for Privacy-Preserving Pedestrian Detection in Thermal Imagery*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2019, vol. 11752, p. 203–213.

[3] A. v. d. B. Paul Voigt, *The EU General Data Protection Regulation (GDPR) - A practical guide*. Cham: Springer International Publishing, 2017.

[4] *General Data Protection Regulation (GDPR) – Official Legal Text*, https://gdpr-info.eu/, (Accessed on 03/02/2022).

[5] M. Fabbri, G. Brasó, G. Maugeri, O. Cetintas, R. Gasparini, A. Ošep, S. Calderara, L. Leal-Taixé, and R. Cucchiara, “Motsynth: How can synthetic data help pedestrian detection and tracking?” in *Proceedings of ICCV*, 2021, pp. 10 849–10 859.

[6] M. Fabbri, F. Lanzi, S. Calderara, A. Palazzi, R. Vezzani, and R. Cucchiara, “Learning to detect and track visible and occluded body joints in a virtual world,” in *Proceedings of ECCV*, 2018.

[7] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of CVPR*, 2020, pp. 2443–2451.

[8] A. Acsintoae, A. Florescu, M.-I. Georgescu, T. Mare, P. Sumedrea, R. T. Ionescu, F. S. Khan, and M. Shah, “Ubnormal: New benchmark for supervised open-set video anomaly detection,” 2021.

[9] R. Blythman, A. Elrasad, E. O’Connell, P. Kielty, M. O’Byrne, M. Moustafa, C. Ryan, and J. Lemley, “Synthetic thermal image generation for human-machine interaction in vehicles,” in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[10] S. Bonderup, J. Olsson, M. Bonderup, and T. B. Moeslund, “Preventing drowning accidents using thermal cameras,” in *Advances in Visual Computing*. Springer International Publishing, 2016, pp. 111–122.

[11] I. Nikolov, J. Liu, and T. Moeslund, “Imitating emergencies: Generating thermal surveillance fall data using low-cost human-like dolls,” *Sensors*, vol. 22, no. 3, 2022.

[12] 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 *Proceedings of NeurIPS*, 2021.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of NIPS*, 2012, pp. 1106–1114.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

[15] V. V. Kniaz, V. A. Knyaz, J. Hladůvka, W. G. Kropatsch, and V. Mizginov, *ThermalGAN: Multimodal Color-to-Thermal Image Translation for Person Reidentification in Multispectral Dataset*. Springer International Publishing, 2019, vol. 11134, p. 606–624.

[16] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, "Synthetic data generation for end-to-end thermal infrared tracking," *IEEE Transactions on Image Processing*, vol. 28, no. 4, p. 1837–1850, 2019.

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

[18] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceeding of ICCV*, 2017, p. 2242–2251.

[19] M. Kieu, L. Berlincioni, L. Galteri, M. Bertini, A. D. Bagdanov, and A. Bimbo, "Robust pedestrian detection in thermal imagery using synthesized images," *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8804–8811, 2021.

[20] C. Pramerdorfer, J. Strohmayer, and M. Kampel, "Sdt: A synthetic multi-modal dataset for person detection and pose classification," in *Proceedings of ICIP*, 2020, pp. 1611–1615.

[21] F. Bongini, L. Berlincioni, M. Bertini, and A. Del Bimbo, *Partially Fake It Till You Make It: Mixing Real and Fake Thermal Images for Improved Object Detection*, 2021, p. 5482–5490.

[22] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 2017, pp. 1–16.

[23] K.-T. Lai, C.-C. Lin, C.-Y. Kang, M.-E. Liao, and M.-S. Chen, "Vivid: Virtual environment for visual deep learning," *Proceedings of the 26th ACM international conference on Multimedia*, 2018.

[24] N. P. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” *Proceeding of IROS*, vol. 3, pp. 2149–2154 vol.3, 2004.

[25] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. X. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, and D. Batra, “Habitat 2.0: Training home assistants to rearrange their habitat,” in *Proceedings of NeurIPS*, 2021, p. 251–266.

[26] J. K. Haas, “A history of the unity game engine,” 2014.

[27] P. Martinez-Gonzalez, S. Oprea, J. A. Castro-Vargas, A. Garcia-Garcia, S. Orts-Escolano, J. A. García-Rodríguez, and M. Vincze, “Unrealrox+: An improved tool for acquiring synthetic data from virtual 3d environments,” *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2021.

[28] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[29] *3DF ZEPHYR - photogrammetry software - 3D models from photos*, (Accessed on 03/01/2022). [Online]. Available: https://www.3dflow.net/3df-zephyr-photogrammetry-software

[30] J. Liu, M. Philipsen, and T. Moeslund, “Supervised versus self-supervised assistant for surveillance of harbor fronts,” in *VISAPP*, 2021, pp. 610–617.

[31] S. R. Nyman, C. Ballinger, J. E. Phillips, and R. Newton, “Characteristics of outdoor falls among older people: a qualitative study,” *BMC Geriatrics*, vol. 13, no. 1, 2013.

[32] W. Li, T. H. M. Keegan, B. Sternfeld, S. Sidney, C. P. Quesenberry, and J. Kelsey, “Outdoor falls among middle-aged and older adults: a neglected public health problem.” *American journal of public health*, vol. 96 7, pp. 1192–200, 2006.

[33] S. Blackman, *Rigging with Mixamo*. Berkeley, CA: Apress, 2014, pp. 565–573.

[34] Hikvision, “Ds-2td2235d-25/50,” https://us.hikvision.com/en/products/more-products/discontinued-products/thermal-camera/thermal-network-bullet-camera-ds, 2015, accessed: 2021-09-27.

[35] Unity Technologies, “Unity Perception package,” https://github.com/Unity-Technologies/com.unity.perception, 2020.

[36] S. K. Bhatia and G. M. Lacy, “Infra-red sensor simulation,” in *I/ITSEC*, 1999, pp. 1–8.

[37] C. Ken and P. Gent, “Image compression and the discrete cosine transform,” *College of the Redwoods, Tech. Rep*, 1998.

[38] O. Shamir and T. Zhang, “Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes,” in *Proceedings of the 30th ICML*, ser. Proceedings of Machine Learning Research, vol. 28. PMLR, 2013, pp. 71–79.

[39] H. Zhang, T. Hu, and J. Zhang, “Surface emissivity of fabric in the 8-14 m waveband,” *J Textile Inst*, vol. 100, 2009.

[40] M. Charlton, S. A. Stanley, Z. Whitman, V. Wenn, T. J. Coats, M. Sims, and J. P. Thompson, “The effect of constitutive pigmentation on the measured emissivity of human skin,” *PLoS ONE*, vol. 15, 2020.

Paper C

Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection

Nicolae-Cătălin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B. Moeslund, Mubarak Shah

The paper has been published in the
Computer Vision and Pattern Recognition (CVPR-Oral), pp. 13566–13576, 2022

The layout has been revised.

Abstract

Anomaly detection is commonly pursued as a one-class classification problem, where models can only learn from normal training samples, while being evaluated on both normal and abnormal test samples. Among the successful approaches for anomaly detection, a distinguished category of methods relies on predicting masked information (e.g.patches, future frames, etc.) and leveraging the reconstruction error with respect to the masked information as an abnormality score. Different from related methods, we propose to integrate the reconstruction-based functionality into a novel self-supervised predictive architectural building block. The proposed self-supervised block is generic and can easily be incorporated into various state-of-the-art anomaly detection methods. Our block starts with a convolutional layer with dilated filters, where the center area of the receptive field is masked. The resulting activation maps are passed through a channel attention module. Our block is equipped with a loss that minimizes the reconstruction error with respect to the masked area in the receptive field. We demonstrate the generality of our block by integrating it into several state-of-the-art frameworks for anomaly detection on image and video, providing empirical evidence that shows considerable performance improvements on MVTec AD, Avenue, and ShanghaiTech. We release our code as open source at: https://github.com/ristea/sspcab.

1 Introduction

Anomaly detection is an important task with a broad set of applications ranging from industrial inspection (finding defects of objects or materials on industrial production lines) [1–8] to public security (detecting abnormal events such as traffic accidents, fights, explosions, etc.) [9–28]. The task is typically framed as a one-class classification (outlier detection) problem, where methods [9, 14, 16, 21, 22, 26, 29–47] learn a familiarity model from normal training samples, labeling unfamiliar examples (outliers) as anomalies, at inference time. Since abnormal samples are available only at test time, supervised learning methods are not directly applicable to anomaly detection. To this end, researchers turned their attention to other directions such as reconstruction-based approaches [4, 5, 7, 13, 19, 21, 31, 35, 37, 42, 44, 48], dictionary learning methods [2, 30, 36, 49–51], distance-based models [3, 14, 22, 32, 40, 41, 43, 52–57], change detection frameworks [20, 58–60], and probabilistic models [6, 29, 33, 38, 39, 61–65].

A distinguished subcategory of reconstruction methods relies on predicting masked information, leveraging the reconstruction error with respect to the masked information as an abnormality score. The masked information can come in different forms, *e.g.*superpixels [5], future frames [35], middle bounding boxes [11], among others. Methods in this subcategory mask some part of the input and employ a deep neural network to predict the missing input information. Different from such methods, we propose to integrate the capability of reconstructing the masked information into a

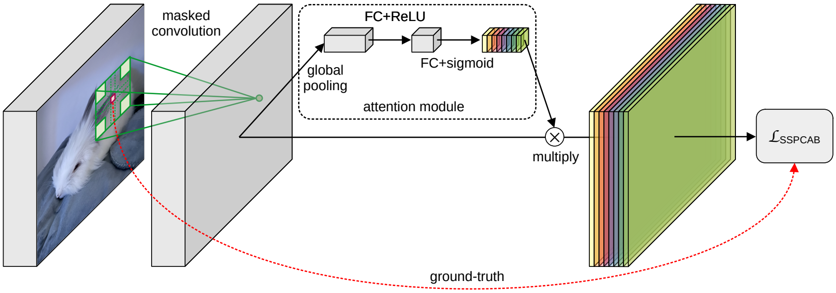


Fig. C.1: Our self-supervised predictive convolutional attentive block (SSPCAB). For each location where the dilated convolutional filter is applied, the block learns to reconstruct the masked area using contextual information. A channel attention module performs feature recalibration by using global information to selectively emphasize or suppress reconstruction maps. Best viewed in color.

neural block. Introducing the reconstruction task at a core architectural level has two important advantages: (i) it allows us to mask information at any layer in a neural network (not only at the input), and (ii) it can be integrated into a wide range of neural architectures, thus being very general.

We design our reconstruction block as a self-supervised predictive block formed of a dilated convolutional layer and a channel attention mechanism. The dilated filters are based on a custom receptive field, where the center area of the kernel is masked. The resulting convolutional activation maps are then passed through a channel attention module [66]. The attention module ensures the block does not simply learn to reconstruct the masked region based on linearly interpolating contextual information. Our block is equipped with a loss that minimizes the reconstruction error between the final activation maps and the masked information. In other words, our block is trained to predict the masked information in a self-supervised manner. Our self-supervised predictive convolutional attentive block (SSPCAB) is illustrated in Figure C.1. For each location where the dilated convolutional filter is applied, the block learns to reconstruct the masked area using contextual information. Meanwhile, the dilation rate becomes a natural way to control the context level (from local to global), as required for the specific application.

We integrate SSPCAB into various state-of-the-art anomaly detection frameworks [12, 17, 21, 35, 67, 68] and conduct comprehensive experiments on the MVTec AD [1], Avenue [36] and ShanghaiTech [37] data sets. Our empirical results show that SSPCAB can bring significant performance improvements, *e.g.*the region-based detection criterion (RBDC) of Liu *et al.* [17] on Avenue increases from 41% to 62% by adding SSPCAB. Moreover, with the help of SSPCAB, we are able to report new state-of-the-art performance levels on Avenue and ShanghaiTech. Additionally, we show extra results on the Avenue data set, indicating that the masked convolutional layer can also increase performance levels, all by itself.

In summary, our contribution is twofold:

- We introduce a novel self-supervised predictive convolutional attentive block that is inherently capable of performing anomaly detection.

- We integrate the block into several state-of-the-art neural models [12, 17, 21, 35, 67, 68] for anomaly detection, showing significant performance improvements across multiple models and benchmarks.

2 Related Work

As anomalies are difficult to anticipate, methods are typically trained only on normal data, while being tested on both normal and abnormal data [21, 31]. Therefore, outlier detection [14, 22, 32, 40, 41] and self-supervised learning [11–13, 17, 18, 21, 67, 68] approaches are extensively used to address the anomaly detection task. Anomaly detection methods can be classified into: dictionary learning methods [2, 30, 36, 49–51], change detection frameworks [20, 58–60], probability-based methods [6, 29, 33, 38, 39, 61–65], distance-based models [3, 14, 22, 32, 40, 41, 43, 52–57], and reconstruction-based methods [4, 5, 7, 13, 19, 21, 31, 35, 37, 42, 44, 48, 68].

Dictionary-based methods learn the normal behavior by constructing a dictionary, where each entry in the dictionary represents a normal pattern. Ren *et al.* [51] extended dictionary learning methods by considering the relation among different entries. Change-detection frameworks detect anomalies by quantifying changes across the video frames, *i.e.*a significant deviation from the immediately preceding event marks the beginning of an abnormal event. After quantifying the change, approaches such as unmasking [59] or ordinal regression [20] can be used to segregate anomalies. Probability-based methods build upon the assumption that anomalies occur in a low probability region. These methods estimate the probability density function (PDF) of the normal data and evaluate the test samples based on the PDF. For example, Mahadevan *et al.* [38] used a Mixture of Dynamic Textures (MDTs) to model the distribution of the spatio-temporal domain, while Rudolph *et al.* [6] employed normalizing flow to represent the normal distribution. Distance-based methods learn a distance function based on the assumption that normal events occur in the close vicinity of the learned feature space, while the abnormal events are far apart from the normal data. For instance, Ramachandra *et al.* [40] employed a Siamese network to learn the distance function. Reconstruction-based methods rely on the assumption that the normal examples can be reconstructed more faithfully from the latent manifold. Our new block belongs to the category of reconstruction-based anomaly detection methods, particularly siding with methods that predict or reconstruct missing (or masked) information [5, 11, 35].

Reconstruction-based methods. In the past few years, reconstruction-based methods became prevalent in anomaly detection. Such methods typically use auto-encoders [31] and generative adversarial networks (GANs) [35], as these neural models enable the learning of powerful reconstruction manifolds via using normal data only. However, the generalization capability of neural networks sometimes leads to reconstructing abnormal frames with low error [9, 12], affecting the discrimination between abnormal and normal frames. To address this issue, researchers have tried to improve the latent manifold by diversifying the architecture and training methodologies. Some works focusing on transforming the architectures include memory-based auto-encoders [9, 17, 21], which memorize the normal prototypes in the training data, thus increasing the discrimination between normal and abnormal samples. Other works remodeled the reconstruction manifold via training the models with pseudo-abnormal samples [12, 68, 69]. The adversarial training proposed in [11] applies gradient ascent for out-of-domain pseudo-abnormal samples and gradient descent for normal data, thus learning a more powerful discriminative manifold for video anomaly detection. Zavrtanik *et al.* [68] created pseudo-abnormal samples by adding random noise patches on normal images for image anomaly detection. Some variants of auto-encoders, such as Variational Auto-Encoders (VAEs), have been proposed in [17, 70] for the anomaly detection task. These works are based on the assumption that VAEs can only reconstruct the normal images. Liu *et al.* [17] used a conditional VAE, conditioning the image prediction on optical flow reconstruction, thus accumulating the error from the optical flow reconstruction task with the image prediction. However, this approach can only be applied to video anomaly detection, due to the presence of motion information in the form of optical flow.

Reconstruction of masked information. A surrogate task for many anomaly detection approaches [4, 27, 35, 71, 72] is to erase some information from the input, while making neural networks predict the erased information. Haselmann *et al.* [71] framed anomaly detection as an inpainting problem, where patches from images are masked randomly, using the pixel-wise reconstruction error of the masked patches for surface anomaly detection. Fei *et al.* [4] proposed the Attribute Restoration Network (AR-Net), which includes an attribute erasing module (AEM) to disorient the model by erasing certain attributes from an image, such as color and orientation. In turn, AR-Net learns to restore the original image and detect anomalies based on the assumption that normal images can be restored properly. The Cloze task [72] is about learning to complete a video when certain frames are removed, being recently employed by Yu *et al.* [27] for anomaly detection. In a similar direction, Georgescu *et al.* [11] proposed middle frame masking as one of the auxiliary tasks for video anomaly detection. Both approaches are based on the assumption that an erased frame can be reconstructed more accurately for regular motion. Future frame prediction [67] utilizes past frames to predict the next frame in the video. The anomaly, in this case, is detected through the prediction error. Another approach based on GANs [73] learns to erase patches from an image, while the discriminator identifies if patches are normal or irregular.

Unlike existing approaches, we are the first to introduce the reconstruction-based functionality as a basic building block for neural architectures. More specifically, we design a novel block based on masked convolution and channel attention to reconstruct a masked part of the convolutional receptive field. As shown in the experiments, our block can be integrated into a multitude of existing anomaly detection frameworks [12, 17, 21, 35, 67, 68], almost always bringing significant performance improvements.

3 Method

Convolutional neural networks (CNNs) [74, 75] are widely used across a broad spectrum of computer vision tasks, also being prevalent in anomaly detection [12, 17, 21, 67, 76]. CNNs are formed of convolutional layers equipped with kernels which learn to activate on discriminative local patterns, in order to solve a desired task. The local features extracted by a convolutional layer are combined into more complex features by the subsequent convolutional layers. From this learning process, a hierarchy of features emerges, ranging from low-level features (corners, edges, etc.) to high-level features (car wheels, bird heads, etc.) [77]. While this hierarchy of features is extremely powerful, CNNs lack the ability to comprehend the global arrangement of such local features, as noted by Sabour *et al.* [78].

In this paper, we introduce a novel self-supervised predictive convolutional attentive block (SSPCAB) that is purposed at learning to predict (or reconstruct) masked information using contextual information. To achieve highly accurate reconstruction results, our block is forced to learn the global structure of the discovered local patterns. Thus, it addresses the issue pointed out in [78], namely the fact that CNNs do not grasp the global arrangement of local features, as they do not generalize to novel viewpoints or affine transformations. To implement this behavior, we design our block as a convolutional layer with dilated masked filters, followed by a channel attention module. The block is equipped with its own loss function, which is aimed at minimizing the reconstruction error between the masked input and the predicted output.

We underline that our design is generic, as SSPCAB can be integrated into just about any CNN architecture, being able to learn to reconstruct masked information, while offering useful features for subsequent neural layers. Although the capability of learning and using global structure might make SSPCAB useful for a wide range of tasks, we conjecture that our block has a natural and direct applicability in anomaly detection, as explained next. When integrated into a CNN trained on normal training data, SSPCAB will learn the global structure of normal examples only. When presented with an abnormal data sample at inference time, our block will likely provide a poor reconstruction. We can thus measure the quality of the reconstruction and employ the result as a way to differentiate between normal and abnormal examples. In Section 4, we provide empirical evidence to support our claims.

SSPCAB is composed of a masked convolutional layer activated by Rectified Linear Units (ReLU) [79], followed by a Squeeze-and-Excitation (SE) module [66]. We next present its components in more details.

Masked convolution. The receptive field of our convolutional filter is depicted in Figure C.2. The learnable parameters of our masked convolution are located in the corners of the receptive field, being denoted by the sub-kernels $\boldsymbol{K}_i \in \mathbb{R}^{k' \times k' \times c}$, $\forall i \in \{1, 2, 3, 4\}$, where $k' \in \mathbb{N}^+$ is a hyperparameter defining the sub-kernel size and c is the number of input channels. Each kernel $\boldsymbol{K}_i$ is located at a distance (dilation

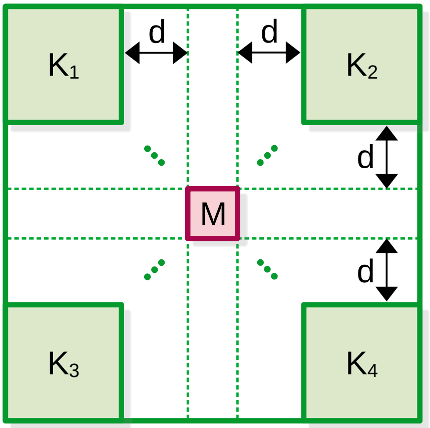


Fig. C.2: Our masked convolutional kernel. The visible area of the receptive field is denoted by the regions $\boldsymbol{K}_i, \forall i \in \{1, 2, 3, 4\}$, while the masked area is denoted by $\boldsymbol{M}$. A dilation factor d controls the local or global nature of the visible information with respect to $\boldsymbol{M}$. Best viewed in color.

rate) $d \in \mathbb{N}^+$ from the masked region in the center of our receptive field, which is denoted by $\boldsymbol{M} \in \mathbb{R}^{1 \times 1 \times c}$. Consequently, the spatial size k of our receptive field can be computed as follows: $k = 2k' + 2d + 1$.

Let $\boldsymbol{X} \in \mathbb{R}^{h \times w \times c}$ be the input tensor of our masked convolutional layer, where c is the number of channels, and h and w are the height and width, respectively. The convolutional operation performed with our custom kernel in a certain location of the input $\boldsymbol{X}$ only considers the input values from the positions where the sub-kernels $\boldsymbol{K}_i$ are located, the other information being ignored. The results of the convolution operations between each $\boldsymbol{K}_i$ and the corresponding inputs are summed into a single number, as if the sub-kernels $\boldsymbol{K}_i$ belong to a single convolutional kernel. The resulting value denotes a prediction located at the same position as $\boldsymbol{M}$. Naturally, applying the convolution with one filter produces a single activation map. Hence, we would only be able to predict one value from the masked vector $\boldsymbol{M}$, at the current location. To predict a value for every channel in $\boldsymbol{M}$, we introduce a number of c masked convolutional filters, each predicting the masked information from a distinct channel. As we aim to learn and predict the reconstruction for every spatial location of the input, we add zero-padding of $k' + d$ pixels around the input and set the stride to 1, such that every pixel in the input is used as masked information. Therefore, the spatial dimension of the output tensor $\boldsymbol{Z}$ is identical to that of the input tensor $\boldsymbol{X}$. Finally, the output tensor is passed through a ReLU activation. We underline that the only configurable hyperparameters of our custom convolutional layer are k' and d.

Channel attention module. Next, the output of the masked convolution is processed by a channel attention module, which computes an attention score for each channel. Knowing that each activation map in $\boldsymbol{Z}$ is predicted by a separate filter in the presence of masked information, we infer that the masked convolution might end up producing activation maps containing disproportionate (uncalibrated) values across channels. Therefore, we aim to exploit the relationships between channels, with the goal of scaling each channel in $\boldsymbol{Z}$ in accordance with the quality of the representations produced by the masked convolutional layer. To this end, we employ the channel attention module proposed by Hu *et al.* [66]. The SE module [66] provides a mechanism that performs adaptive recalibration of channel-wise feature responses. Through this mechanism, it can learn to use global information to selectively emphasize or suppress reconstruction maps, as necessary. Another motivation to use attention is to increase the modeling capacity of SSPCAB and enable a non-linear processing between the input and output of our block.

Formally, the channel attention block reduces $\boldsymbol{Z}$ to a vector $\boldsymbol{z} \in \mathbb{R}^c$ through a global pooling performed on each channel. Subsequently, the vector of scale factors $\boldsymbol{s} \in \mathbb{R}^c$ is computed as follows:

$$\boldsymbol{s} = \sigma\left(\boldsymbol{W}_2 \cdot \delta\left(\boldsymbol{W}_1 \cdot \boldsymbol{z}\right)\right), \tag{C.1}$$

where σ is the sigmoid activation, δ is the ReLU activation, and $\boldsymbol{W}_1 \in \mathbb{R}^{\frac{c}{r} \times c}$ and $\boldsymbol{W}_2 \in \mathbb{R}^{c \times \frac{c}{r}}$ represent the weight matrices of two consecutive fully connected (FC) layers, respectively. The first FC layer consists of $\frac{c}{r}$ neurons, squeezing the information by a reduction ratio of r.

Next, the vector $\boldsymbol{s}$ is replicated in the spatial dimension, generating a tensor $\boldsymbol{S}$ of the same size as $\boldsymbol{Z}$. Our last step is the element-wise multiplication between $\boldsymbol{S}$ and $\boldsymbol{Z}$, producing the final tensor $\hat{\boldsymbol{X}} \in \mathbb{R}^{h \times w \times c}$ containing recalibrated features maps.

Reconstruction loss. We add a self-supervised task consisting of reconstructing the masked region inside our convolutional receptive field, for every location where the masked filters are applied. To this end, our block should learn to provide the corresponding reconstructions as the output $\hat{\boldsymbol{X}}$. Let G denote the SSPCAB function. We define the self-supervised reconstruction loss as the mean squared error (MSE) between the input and the output, as follows:

$$\mathcal{L}_{\text{SSPCAB}}(G, \boldsymbol{X}) = \left(G(\boldsymbol{X}) - \boldsymbol{X}\right)^2 = \left(\hat{\boldsymbol{X}} - \boldsymbol{X}\right)^2. \tag{C.2}$$

When integrating SSPCAB into a neural model F having its own loss function $\mathcal{L}_F$, our loss can simply be added to the respective loss, resulting in a new loss function that comprises both terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_F + \lambda \cdot \mathcal{L}_{\text{SSPCAB}}, \tag{C.3}$$

where $\lambda \in \mathbb{R}^+$ is a hyperparameter that controls the importance of our loss with respect to $\mathcal{L}_F$. We adopt this procedure when incorporating SSPCAB into various neural architectures during our experiments.

4 Experiments and Results

4.1 Data Sets

MVTec AD. The MVTec AD [1] data set is a standard benchmark for evaluating anomaly detection methods on industrial inspection images. It contains images from 10 object categories and 5 texture categories, having 15 categories in total. There are 3629 defect-free training images and 1725 test images with or without anomalies.
Avenue. The CHUK Avenue [36] data set is a popular benchmark for video anomaly detection. It contains 16 training and 21 test videos. The anomalies are present only at inference time and include people throwing papers, running, dancing, loitering, and walking in the wrong direction.
ShanghaiTech. The ShanghaiTech [37] benchmark is one of the largest data sets for video anomaly detection. It is formed of 330 training and 107 test videos. As for Avenue, the training videos contain only normal samples, but the test videos can contain both normal and abnormal events. Some examples of anomalies are: people fighting, stealing, chasing, jumping, and riding bike or skating in pedestrian zones.

4.2 Evaluation Metrics

Image anomaly detection. On MVTec AD, we evaluate methods in terms of the average precision (AP) and the area under the receiver operating characteristic curve (AUROC). The ROC curve is obtained by plotting the true positive rate (TPR) versus the false positive rate (FPR). We consider both localization and detection performance rates. For the detection task, the TPR and FPR values are computed at the image level, *i.e.*TPR is the percentage of anomalous images that are correctly classified, while FPR is the percentage of normal images mistakenly classified as anomalous. For the localization (segmentation) task, TPR is the percentage of abnormal pixels that are correctly classified, whereas FPR is the percentage of normal pixels wrongly classified as anomalous. To determine the segmentation threshold for each method, we follow the approach described in [1].
Video anomaly detection. We evaluate abnormal event detection methods in terms of the area under the curve (AUC), which is computed by marking a frame as abnormal if at least one pixel inside the frame is abnormal. Following [12], we report both the macro and micro AUC scores. The micro AUC is computed after concatenating all frames from the entire test set, while the macro AUC is the average of the AUC scores on individual videos. The frame-level AUC can be an unreliable evaluation measure, as it may fail to evaluate the localization of anomalies [22]. Therefore, we also evaluate models in terms of the region-based detection criterion (RBDC) and track-based detection criterion (TBDC), as proposed by Ramachandra *et al.* [22]. RBDC takes each detected region into consideration, marking a detected region as *true positive* if the Intersection-over-Union with the ground-truth region is greater than a threshold α. TBDC measures whether abnormal regions are accurately tracked across time. It con-

Method	Loss type	d	k'	r	Attention type	AUC		RBDC	TBDC
						Micro	Macro		
Plain auto-encoder	-	-	-	-	-	80.0	83.4	49.98	51.69
	MAE	0	1	-	-	83.3	84.1	47.46	52.11
		1	1	-	-	83.9	84.6	49.05	52.21
		2	1	-	-	83.2	84.3	48.56	52.03
	MSE	0	1	-	-	83.6	84.2	47.86	52.21
		1	1	-	-	84.2	84.9	49.22	52.29
		2	1	-	-	83.6	84.3	48.44	51.98
	MSE	0	2	-	-	83.7	84.0	47.41	53.02
		1	2	-	-	84.0	85.1	48.22	51.84
		2	2	-	-	82.7	83.1	46.94	50.22
	MSE	0	3	-	-	82.6	83.7	48.28	51.91
		1	3	-	-	82.9	84.7	48.13	52.07
		2	3	-	-	83.1	83.8	47.13	49.96
	MSE	1	1	8	CA	**85.9**	**85.6**	53.81	**56.33**
		1	1	-	SA	84.3	84.4	53.31	53.41
		1	1	8	CA+SA	85.7	85.6	**53.98**	54.11
	MSE	1	1	4	CA	85.6	85.3	53.83	55.99
		1	1	16	CA	84.4	84.9	53.28	54.37

Table C.1: Micro AUC, macro AUC, RBDC and TBDC scores (in %) obtained on the Avenue data set with different hyperparameter configurations, *i.e.*kernel size (k'), dilation rate (d), reduction ratio (r), loss type, and attention type, for our SSPCAB. Results are obtained by introducing SSPCAB into a plain auto-encoder that follows the basic architecture designed by Georgescu *et al.* [12]. Best results are highlighted in bold.

siders a detected track as *true positive* if the number of detections in a track is greater than a threshold β. Following [12, 22], we set $\alpha = 0.1$ and $\beta = 0.1$.

4.3 Implementation Choices and Tuning

For the methods [12, 17, 21, 35, 67, 68] chosen to serve as underlying models for SSPCAB, we use the official code from the repositories provided by the corresponding authors, inheriting the hyperparameters, *e.g.*the number of epochs and learning rate, from each method. Unless specified otherwise, we replace the penultimate convolutional layer with SSPCAB in all underlying models.

In a set of preliminary trials with a basic auto-encoder on Avenue, we tuned the hyperparameter λ from Eq. (C.3), representing the weight of the SSPCAB reconstruction error, considering values between 0.1 and 1, at a step of 0.1. Based on these preliminary trials, we decided to use $\lambda = 0.1$ across all models and data sets. However, we observed that $\lambda = 0.1$ gives a higher than necessary magnitude to our loss for the framework of Liu *et al.* [17]. Hence, for Liu *et al.* [17], we reduced λ to 0.01.

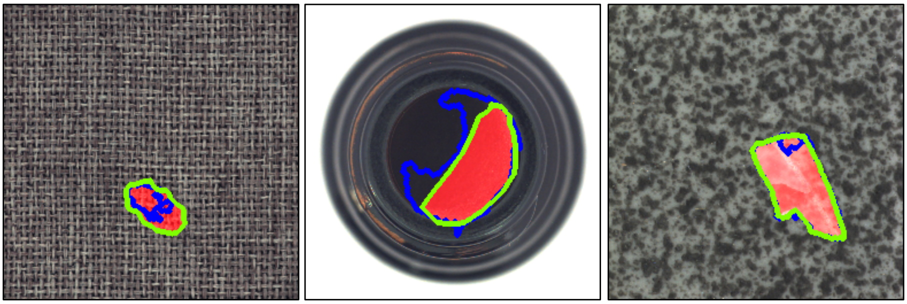


Fig. C.3: Anomaly localization examples of DRAEM [68] (blue) versus DRAEM+SSPCAB (green) on MVTec AD. The ground-truth anomalies are marked with a red mask. Best viewed in color.

4.4 Preliminary Results

We performed preliminary experiments on Avenue to decide the hyperparameters of our masked convolution, *i.e.*the kernel size k' and dilation rate d. We consider values in $\{1, 2, 3\}$ for k', and values in $\{0, 1, 2\}$ for d. In addition, we consider two alternative loss functions, namely the Mean Absolute Error (MAE) and Mean Squared Error (MSE), and several types of attention to be added after the masked convolution, namely channel attention (CA), spatial attention (SA), and both (CA+SA).

For the preliminary experiments, we take the appearance convolutional auto-encoder from [12] as our baseline, stripping out the additional components such as optical flow, skip connections, adversarial training, mask reconstruction and binary classifiers. Our aim is to test various SSPCAB configurations on top of a basic architecture, without trying to overfit the configuration to a specific framework, such as that of Georgescu *et al.* [12]. To this end, we decided to remove the aforementioned components, thus using only a plain auto-encoder in our preliminary experiments.

The preliminary results are presented in Table C.1. Upon adding the masked convolutional layer based on the MAE loss on top of the basic architecture, we observe significant performance gains, especially for $k' = 1$ and $d = 1$. The performance further increases when we replace the MAE loss function with MSE. We performed extensive experiments with different combinations of k' and d, obtaining better results with $k' = 1$ and $d = 1$. We therefore decided to fix the loss to MSE, the sub-kernel size k' to 1, and the dilation rate d to 1, for all subsequent experiments. Next, we introduced various attention modules after our masked convolution. Among the considered attention modules, we observe that channel attention is the one that better compliments our masked convolutional layer, providing the highest performance gains for three of the metrics: 5.9% for the micro AUC, 2.2% for the macro AUC, and 4.6% for TBDC. Accordingly, we selected the channel attention module for the remaining experiments. Upon choosing to use channel attention, we test additional reduction rates ($r = 4$ and $r = 16$), without observing any improvements. As such, we keep the reduction rate of the SE module to $r = 8$, whenever we integrate SSPCAB into a neural model.

	Class	Localization				Detection					
		DRAEM [68]				DRAEM [68]		CutPaste [67]			
								3-way		Ensemble	
		AUROC		AP		AUROC		AUROC		AUROC	
			+SSPCAB		+SSPCAB		+SSPCAB		+SSPCAB		+SSPCAB
Texture	Carpet	**95.5**	95.0	53.5	**59.4**	97.0	**98.2**	**93.1**	90.7	93.9	**96.8**
	Grid	**99.7**	99.5	**65.7**	61.1	99.9	**100.0**	99.9	99.9	**100.0**	99.9
	Leather	98.6	**99.5**	75.3	**76.0**	100.0	100.0	100.0	100.0	100.0	100.0
	Tile	99.2	**99.3**	92.3	**95.0**	99.6	**100.0**	93.4	**94.0**	94.6	**95.0**
	Wood	96.4	**96.8**	77.7	77.1	99.1	**99.5**	98.6	**99.2**	99.1	99.1
Object	Bottle	**99.1**	98.8	86.5	**87.9**	**99.2**	98.4	98.3	**98.6**	98.2	**99.1**
	Cable	94.7	**96.0**	52.4	**57.2**	91.8	**96.9**	80.6	**82.9**	81.2	**83.6**
	Capsule	**94.3**	93.1	49.4	**50.2**	98.5	**99.3**	96.2	**98.1**	**98.2**	97.6
	Hazelnut	99.7	**99.8**	92.9	92.6	100.0	100.0	97.3	**98.3**	98.3	**98.4**
	Metal Nut	**99.5**	98.9	96.3	**98.1**	98.7	**100.0**	99.3	**99.6**	99.9	99.9
	Pill	**97.6**	97.5	48.5	**52.4**	98.9	**99.8**	92.4	**95.3**	94.9	**96.6**
	Screw	97.6	**99.8**	58.2	**72.0**	93.9	**97.9**	86.3	**90.8**	88.7	**90.8**
	Toothbrush	98.1	98.1	44.7	**51.0**	100.0	100.0	98.3	**98.8**	99.4	**99.6**
	Transistor	**90.9**	87.0	**50.7**	48.0	**93.1**	92.9	95.5	**96.5**	96.1	**97.3**
	Zipper	98.8	**99.0**	**81.5**	77.1	100.0	100.0	**99.4**	99.1	99.9	99.9
	Overall	**97.3**	97.2	68.4	**69.9**	98.0	**98.9**	95.2	**96.1**	96.1	**96.9**

Table C.2: Localization AUROC/AP and detection AUROC (in %) of state-of-the-art methods on MVTec AD, before and after adding SSPCAB. The best result for each before-versus-after pair is highlighted in bold.

4.5 Anomaly Detection in Images

Baselines. We choose two recent models for image anomaly detection, *i.e.***CutPaste** [67] and **DRAEM** [68].

Li *et al.* [67] proposed *CutPaste*, a simple data augmentation technique that cuts a patch from an image and pastes it to a random location. The CutPaste architecture is built on top of GradCAM [80]. The model is based on a self-supervised 3-way classification task, learning to classify samples into normal, CutPaste and CutPaste-Scar, where a scar is a long and thin mark of a random color. Li *et al.* [67] also used an ensemble of five 3-way CutPaste models trained with different random seeds to improve results.

Zavrtanik *et al.* [68] introduced *DRAEM*, a method based on a dual auto-encoder for anomaly detection and localization on MVTec AD. We introduce SSPCAB into both the localization and detection networks.

Results. We present the results on MVTec AD in Table C.2. Considering the detection results, we observe that SSPCAB brings consistent performance improvements on most categories for both CutPaste [67] and DRAEM [68]. Moreover, the overall

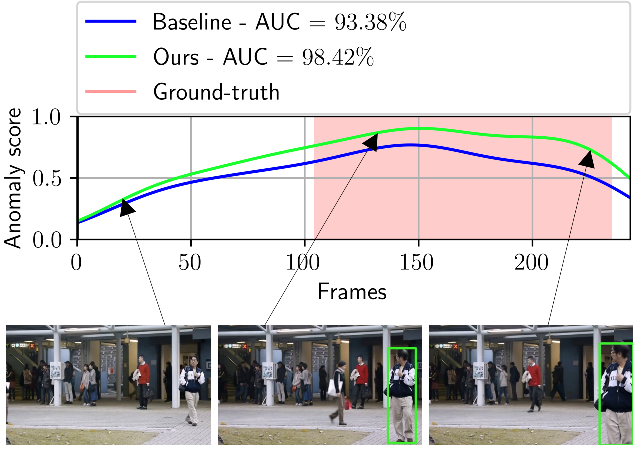


Fig. C.4: Frame-level anomaly scores for Liu *et al.* [35] before (baseline) and after (ours) integrating SSPCAB, for test video 18 from Avenue. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

performance gains in terms of detection AUROC are close to 1%, regardless of the underlying model. Given that the baselines are already very good, we consider the improvements brought by SSPCAB as noteworthy.

Considering the localization results, it seems that SSPCAB is not able to improve the overall AUROC score of DRAEM [68]. However, the more challenging AP metric tells a different story. Indeed, SSPCAB increases the overall AP of DRAEM [68] by 1.5%, from 68.4% to 69.9%.

In Figure C.3, we illustrate a few anomaly localization examples where SSPCAB introduces significant changes to the anomaly localization contours of DRAEM [68], showing a higher overlap with the ground-truth anomalies. We believe that these improvements are a direct effect induced by the reconstruction errors produced by our novel block. We provide more anomaly detection examples in the supplementary.

4.6 Abnormal Event Detection in Video

Baselines. We choose four recently introduced methods [12, 17, 21, 35] attaining state-of-the-art performance levels in video anomaly detection, as candidates for integrating SSPCAB. We first reproduce the results using the official implementations provided by the corresponding authors [12, 17, 21, 35]. We refrain from making any modification to the hyperparameters of the chosen baselines. Despite using the unmodified code from the official repositories, we were not able to exactly reproduce the results of Liu *et al.* [17] and Park *et al.* [21], but our numbers are very close. As we add SSPCAB into the reproduced models, we consider the reproduced results as reference. We underline that, for Georgescu *et al.* [12], we integrate SSPCAB into the auto-encoders, not in the binary classifiers. We report RBDC and TBDC results when-

Method	Avenue				ShanghaiTech			
	AUC		RBDC	TBDC	AUC		RBDC	TBDC
	Micro	Macro			Micro	Macro		
Liu *et al.* [60]	84.4	-	-	-	-	-	-	-
Sultani *et al.* [81]	-	-	-	-	-	76.5	-	-
Lee *et al.* [82]	87.2		-	-	76.2		-	-
Ionescu *et al.* [14]	88.9	-	-	-	-	-	-	-
Nguyen *et al.* [19]	86.9	-	-	-	-	-	-	-
Ionescu *et al.* [32]	87.4	90.4	15.77	27.01	78.7	84.9	20.65	44.54
Wu *et al.* [26]	86.6		-	-	-	-	-	-
Lee *et al.* [16]	90.0		-	-	-	-	-	-
Yu *et al.* [27]	89.6	-	-	-	74.8	-	-	-
Ramachandra *et al.* [22]	72.0		35.80	80.90	-	-	-	-
Ramachandra *et al.* [40]	87.2		41.20	78.60	-	-	-	-
Tang *et al.* [44]	85.1		-	-	73.0		-	-
Dong *et al.* [9]	84.9		-	-	73.7		-	-
Doshi *et al.* [10]	86.4		-	-	71.6		-	-
Sun *et al.* [24]	89.6		-	-	74.7		-	-
Wang *et al.* [25]	87.0		-	-	79.3		-	-
Astrid *et al.* [69]	84.7	-	-	-	73.7	-	-	-
Astrid *et al.* [83]	87.1	-	-	-	75.9	-	-	-
Georgescu *et al.* [11]	91.5	92.8	57.00	58.30	82.4	90.2	42.80	83.90
Liu *et al.* [35]	85.1	81.7	19.59	56.01	72.8	80.6	17.03	54.23
Liu *et al.* [35] + SSPCAB	**87.3**	**84.5**	**20.13**	**62.30**	**74.5**	**82.9**	**18.51**	**60.22**
Park *et al.* [21]	82.8	86.8	-	-	68.3	79.7	-	-
Park *et al.* [21] + SSPCAB	**84.8**	**88.6**	-	-	**69.8**	**80.2**	-	-
Liu *et al.* [17]	89.9	**93.5**	41.05	86.18	74.2	83.2	44.41	83.86
Liu *et al.* [17] + SSPCAB	**90.9**	92.2	**62.27**	**89.28**	**75.5**	**83.7**	**45.45**	**84.50**
Georgescu *et al.* [12]	92.3	90.4	65.05	**66.85**	82.7	89.3	**41.34**	78.79
Georgescu *et al.* [12] + SSPCAB	**92.9**	**91.9**	**65.99**	64.91	**83.6**	**89.5**	40.55	**83.46**

Table C.3: Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech. Among the existing models, we select four models [12, 17, 21, 35] to show results before and after including SSPCAB. The best result for each before-versus-after pair is highlighted in bold. The top score for each metric is shown in red.

ever possible, computing the scores using the implementation provided by Georgescu *et al.* [12].

Results. We report the results on Avenue and ShanghaiTech in Table C.3. First, we observe that the inclusion of SSPCAB in the framework of Liu *et al.* [35] brings consistent improvements over all metrics on both benchmarks. Similarly, we observe consistent performance gains when integrating SSPCAB into the model of Park *et al.* [21]. We note that the method of Park *et al.* [21] does not produce anomaly localization results, preventing us from computing the RBDC and TBDC scores for their method. SSPCAB also brings consistent improvements for Liu *et al.* [17], the only exception being the macro AUC on Avenue. For this baseline [17], we observe a remarkable increase of 21.22% in terms of the RBDC score on Avenue. Finally, we notice that SSPCAB also improves the performance of the approach proposed by Georgescu *et al.* [12] for almost all metrics, the exceptions being the TBDC on Avenue and RBDC on ShanghaiTech. In summary, we conclude that integrating SSPCAB is beneficial, regardless of the underlying model. Moreover, due to the integration of SSPCAB, we are able to report new state-of-the-art results on Avenue and ShanghaiTech, for several metrics.

In Figure C.4, we compare the frame-level anomaly scores on test video 18 from Avenue, before and after integrating SSPCAB into the method of Liu *et al.* [35]. On this video, SSPCAB increases the AUC by more than 5%. We observe that the approach based on SSPCAB can precisely localize and detect the abnormal event (*person walking in the wrong direction*). We provide more anomaly detection examples in the supplementary.

5 Conclusion

In this paper, we introduced SSPCAB, a novel neural block composed of a masked convolutional layer and a channel attention module, which predicts a masked region in the convolutional receptive field. Our neural block is trained in a self-supervised manner, via a reconstruction loss of its own. To show the benefit of using SSPCAB in anomaly detection, we integrated our block into a series of image and video anomaly detection methods [12, 17, 21, 35, 67, 68]. Our empirical results indicate that SSPCAB brings performance improvements in almost all cases. The preliminary results show that both the masked convolution and the channel attention contribute to the performance gains. Furthermore, with the help of SSPCAB, we are able to obtain new state-of-the-art levels on Avenue and ShanghaiTech. We consider this as a major achievement.

In future work, we aim to extend SSPCAB by replacing the masked convolution with a masked 3D convolution. In addition, we aim to consider other application domains besides anomaly detection.

Acknowledgments

The research leading to these results has received funding from the EEA Grants 2014-2021, under Project contract no. EEA-RO-NO-2018-0496. This work has also been funded by the Milestone Research Programme at AAU, SecurifAI, and the Romanian Young Academy, which is funded by Stiftung Mercator and the Alexander von Humboldt Foundation for the period 2020-2022.

C.A Supplementary Material

In the main article, we mention that we generally replace the penultimate convolutional layer with SSPCAB in underlying models [12, 17, 21, 35, 67, 68]. Ideally, for optimal performance gains, the integration place and the number of SSPCAB modules should be tuned on a validation set for each framework. However, anomaly detection data sets do not have a validation set and there is no way to obtain one from the training set, as the training contains only normal examples. In this context, to fairly demonstrate the generality and utility of SSPCAB, we only used a single configuration (one block, closer to the output) across all existing frameworks. However, adding more modules could be beneficial. To test various configurations, we perform an ablation study on the number of SSPCAB modules and the places where these modules can be integrated in a plain auto-encoder. In Table C.4, we present the corresponding experiments on the Avenue data set. *We observe that SSPCAB improves the results, regardless of the place of integration or the number of blocks.* The improvements seem larger when SSPCAB is integrated closer to the output. Integrating more blocks can sometimes help.

	Location of SSPCAB			AUC		RBDC	TBDC
	Early	Middle	Late	Micro	Macro		
Plain auto-encoder				80.0	83.4	49.98	51.69
	✓			81.1	83.6	50.86	52.44
		✓		84.2	85.0	52.73	54.02
			✓	85.9	85.6	53.81	56.33
	✓	✓		82.7	83.8	50.54	52.70
	✓		✓	83.2	84.1	52.33	53.01
		✓	✓	**86.1**	**85.7**	**54.03**	56.07
	✓	✓	✓	85.3	85.4	53.11	**56.64**

Table C.4: Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) on Avenue, while integrating SSPCAB into an auto-encoder, at different locations. SSPCAB improves the results regardless of the integration place or the number of blocks. The option highlighted in red is used throughout the experiments presented in the main article. Best results are highlighted in bold.

Size of M	AUC		RBDC	TBDC
	Micro	Macro		
	80.0	83.4	49.98	51.69
1×1	85.9	85.6	53.81	56.33
3×3	85.9	85.5	53.93	56.31

Table C.5: Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) on Avenue, while varying the size of the masked kernel M.

Another hyperparameter that could be tuned is the size of the masked kernel M. In our experiments, we kept M to a size of 1×1 for simplicity and speed. To study the effect of increasing the size of M, we have tested the size of 3×3 with the plain auto-encoder on Avenue. We report the corresponding results in Table C.5. When comparing the results with masked kernels of 1×1 or 3×3 components, we do not observe significant differences.

An additional aspect that can suffer multiple reconfigurations, given a validation set, is the pattern of the proposed kernel. In our experiments, we tried a simple pattern where the mask is placed in the center and the reception field is connected to the four corner sub-kernels denoted by K_i, $\forall i \in \{1,2,3,4\}$. We designed this pattern while trying to extrapolate the idea from middle frame prediction (which was shown to provide somewhat better results than future frame prediction) to a 2D kernel. Of

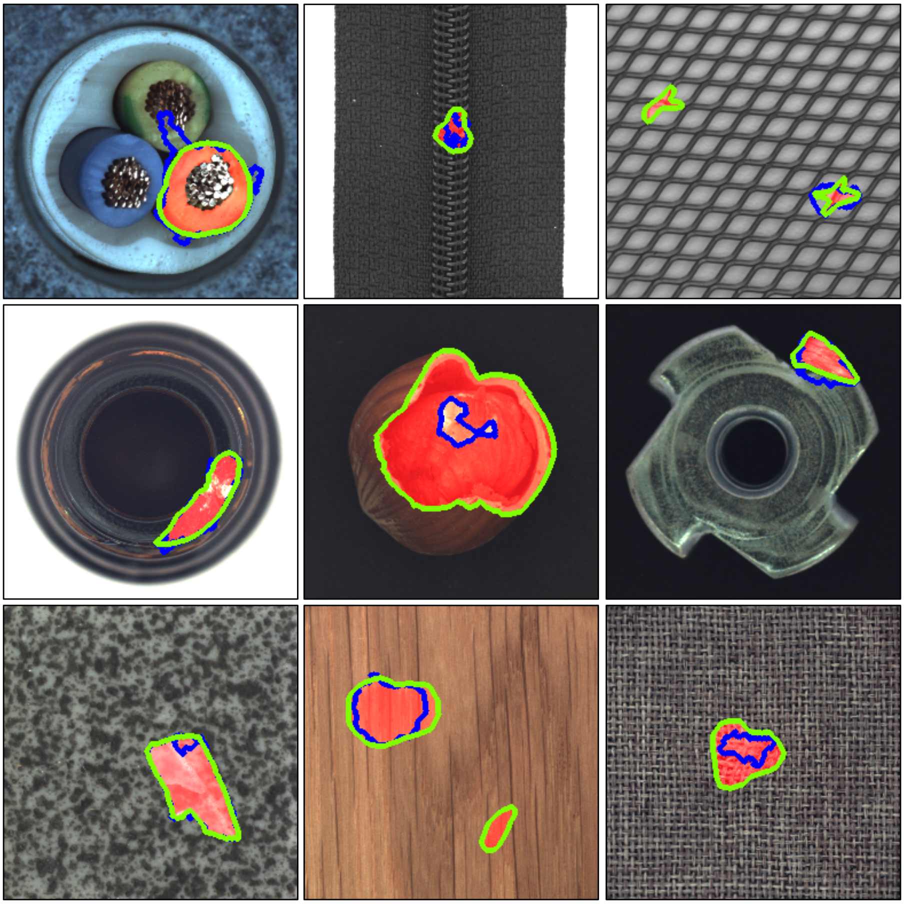


Fig. C.5: Additional anomaly localization examples of DRAEM [68] (blue) versus DRAEM+SSPCAB (green) on MVTec AD. The ground-truth anomalies are marked with a red mask. Best viewed in color.

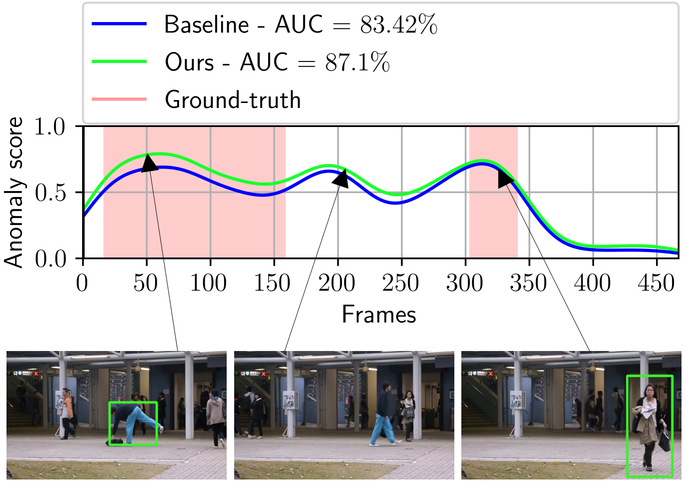


Fig. C.6: Frame-level anomaly scores for Liu *et al.* [35] before (baseline) and after (ours) integrating SSPCAB, for test video 10 from Avenue. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

course, other patterns are possible and are likely to work equally well.

C.B Qualitative Anomaly Detection Results

Anomaly detection in images. In Figure C.5, we present additional qualitative results produced by DRAEM [68] on the MVTec AD benchmark. The displayed examples illustrate the benefit of integrating SSPCAB, which is much better at segmenting the anomalies compared to the baseline DRAEM. We show improvements in terms of the pixel-level annotation for both objects and textures.

Anomaly detection in videos. In Figure C.6, we show a comparison of the frame-level anomaly scores on test video 10 from the Avenue data set, before and after integrating SSPCAB into the method of Liu *et al.* [35]. On this video, SSPCAB increases the AUC by nearly 4%. After introducing SSPCAB, we observe higher frame-level anomaly scores for the first abnormal event. The anomaly localization results depict *a person throwing a backpack* and *a person walking in the wrong direction*.

In Figures C.7 and C.8, we illustrate similar comparisons for test videos 01_0054 and 01_0130 from the ShanghaiTech data set, before and after adding SSPCAB into the framework of Georgescu *et al.* [12]. For test video 01_0054, SSPCAB increases the AUC by more than 10%. For test video 01_0130, the baseline framework seems to detect the abnormal event too early, but SSPCAB seems capable of shifting the detection towards the correct moment. As a result, SSPCAB increases the frame-level AUC score by almost 6%. We observe a similar AUC improvement from SSPCAB in Figure C.9, where we compare the frame-level anomaly scores on test video 07_0047

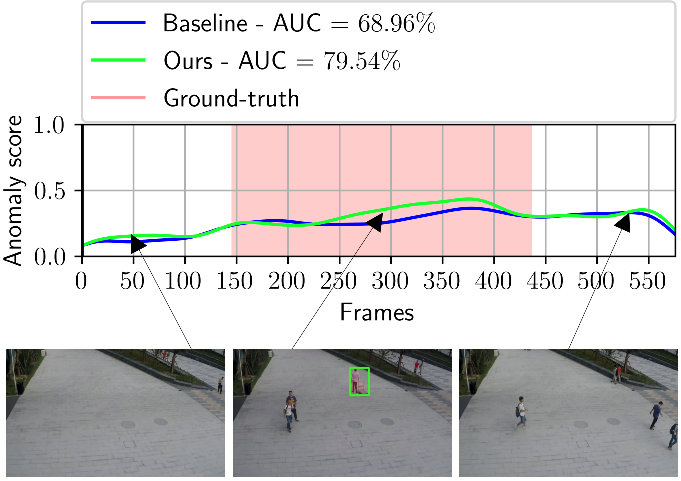


Fig. C.7: Frame-level anomaly scores for Georgescu *et al.* [12] before (baseline) and after (ours) integrating SSPCAB, for test video 01_0054 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

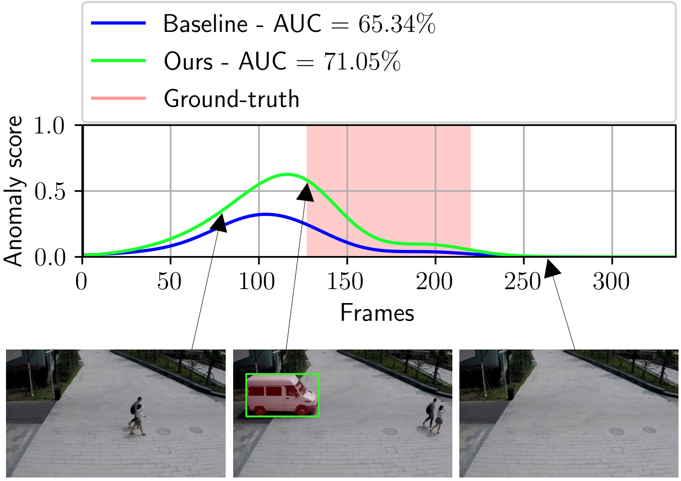


Fig. C.8: Frame-level anomaly scores for Georgescu *et al.* [12] before (baseline) and after (ours) integrating SSPCAB, for test video 01_0130 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

from the ShanghaiTech data set. For this video, we underline that the frame-level scores are visibly more correlated to the ground-truth anomalies. Moreover, in all three ShanghaiTech videos, we observe that the approach based on SSPCAB can pre-

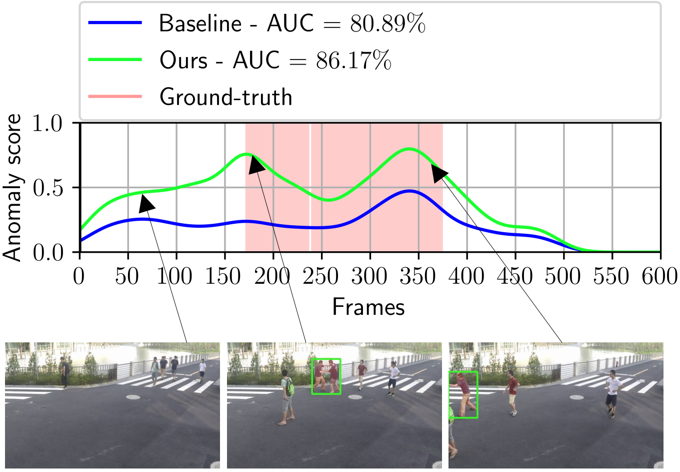


Fig. C.9: Frame-level anomaly scores for Georgescu *et al.* [12] before (baseline) and after (ours) integrating SSPCAB, for test video 07_0047 from ShanghaiTech. Anomaly localization results correspond to the model based on SSPCAB. Best viewed in color.

Method	Time (ms)		Relative (%)
	Baseline	+SSPCAB	
Liu *et al.* [35]	2.1	2.4	14.2
Georgescu *et al.* [12]	1.5	1.7	13.3

Table C.6: Inference times (in milliseconds) and relative time expansions (in %) for two frameworks [12, 35], before and after integrating SSPCAB. The running times are measured on an Nvidia GeForce GTX 3090 GPU with 24 GB of VRAM.

cisely localize and detect the abnormal events (*person pulling a lever cart*, *car inside pedestrian area*, *people fighting*, *people running*).

C.C Inference Time

Regardless of the underlying framework [12, 17, 21, 35, 67, 68], we add only one instance of SSPCAB, usually replacing the penultimate convolutional layer. As such, we expect the running time to increase. To assess the amount of extra time added by SSPCAB, we present the running times before and after integrating SSPCAB into two state-of-the-art frameworks [12, 35] in Table C.6. The reported times show time expansions lower than 0.3 ms for both frameworks. Hence, we consider that the accuracy gains brought by SSPCAB outweigh the marginal running time expansions observed in Table C.6.

C.D Discussion

Although SSPCAB belongs to an existing family of anomaly detection methods, *i.e*. reconstruction-based frameworks [4, 5, 7, 13, 19, 21, 31, 35, 37, 42, 44, 48], we would like to underline that we are the first to integrate the reconstruction functionality at the block level. Unlike other reconstruction approaches, our contribution is more flexible, as it can be integrated in existing and future reconstruction methods. Moreover, SSPCAB can also be used to introduce reconstruction-based anomaly detection in other frameworks, which do not rely on reconstruction. We thus believe that our generic and effective approach will help ease future research in anomaly detection.

An important aspect that must be noted is that, due to the masked convolution, our block will not reconstruct the input exactly. Except for the degenerate case where the input is constant, this scenario should not occur in the real world, which means that the reconstruction performed by SSPCAB is not trivial. However, our foremost intuition about the usefulness of SSPCAB is different: our block provides a better reconstruction for normal convolutional features than for abnormal convolutional features. If the features representing normal versus abnormal examples are different at any layer of a neural architecture, it should result in greater differences at the final output of the architecture. This idea is also supported by the experiments presented in Table C.4.

Further looking at the results shown in Table C.4, we observe that SSPCAB does not bring significant gains when the block is placed near the input. We aim to further investigate this limitation in future work. Aside from this small issue, we did not observe other limitations of SSPCAB during our experiments.

References

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” in *Proceedings of CVPR*, 2019, pp. 9592–9600.

[2] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, “Defect Detection in SEM Images of Nanofibrous Materials,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 551–561, 2017.

[3] T. Defard, A. Setkov, A. Loesch, and R. Audigier, “PaDiM: A patch distribution modeling framework for anomaly detection and localization,” in *Proceedings of ICPR*, 2021, pp. 475–489.

[4] Y. Fei, C. Huang, C. Jinkun, M. Li, Y. Zhang, and C. Lu, “Attribute Restoration Framework for Anomaly Detection,” *IEEE Transactions on Multimedia*, pp. 1–1, 2020.

[5] Z. Li, N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, “Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection,” in *Proceedings of BMVC*, 2020.

[6] M. Rudolph, B. Wandt, and B. Rosenhahn, “Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows,” in *Proceedings of WACV*, 2021, pp. 1907–1916.

[7] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, “Multiresolution Knowledge Distillation for Anomaly Detection,” in *Proceedings of CVPR*, 2021, pp. 14 902–14 912.

[8] J. Yi and S. Yoon, “Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation,” in *Proceedings of ACCV*, 2020, pp. 375–390.

[9] F. Dong, Y. Zhang, and X. Nie, “Dual Discriminator Generative Adversarial Network for Video Anomaly Detection,” *IEEE Access*, vol. 8, pp. 88 170–88 176, 2020.

[10] K. Doshi and Y. Yilmaz, “Any-Shot Sequential Anomaly Detection in Surveillance Videos,” in *Proceedings of CVPRW*, 2020, pp. 934–935.

[11] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “Anomaly Detection in Video via Self-Supervised and Multi-Task Learning,” in *Proceedings of CVPR*, 2021, pp. 12 742–12 752.

[12] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[13] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection," in *Proceedings of ICCV*, 2019, pp. 1705–1714.

[14] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, "Detecting abnormal events in video using Narrowed Normality Clusters," in *Proceedings of WACV*, 2019, pp. 1951–1960.

[15] X. Ji, B. Li, and Y. Zhu, "TAM-Net: Temporal Enhanced Appearance-to-Motion Generative Network for Video Anomaly Detection," in *Proceedings of IJCNN*, 2020, pp. 1–8.

[16] S. Lee, H. G. Kim, and Y. M. Ro, "BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection," *IEEE Transactions on Image Processing*, vol. 29, pp. 2395–2408, 2019.

[17] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction," in *Proceedings of ICCV*, 2021, pp. 13 588–13 597.

[18] Y. Lu, F. Yu, M. Kumar, K. Reddy, and Y. Wang, "Few-Shot Scene-Adaptive Anomaly Detection," in *Proceedings of ECCV*, 2020, pp. 125–141.

[19] T.-N. Nguyen and J. Meunier, "Anomaly Detection in Video Sequence With Appearance-Motion Correspondence," in *Proceedings of ICCV*, 2019, pp. 1273–1283.

[20] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection," in *Proceedings of CVPR*, 2020, pp. 12 173–12 182.

[21] H. Park, J. Noh, and B. Ham, "Learning Memory-guided Normality for Anomaly Detection," in *Proceedings of CVPR*, 2020, pp. 14 372–14 381.

[22] B. Ramachandra and M. Jones, "Street Scene: A new dataset and evaluation protocol for video anomaly detection," in *Proceedings of WACV*, 2020, pp. 2569–2578.

[23] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, "A Survey of Single-Scene Video Anomaly Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[24] C. Sun, Y. Jia, Y. Hu, and Y. Wu, "Scene-Aware Context Reasoning for Unsupervised Abnormal Event Detection in Videos," in *Proceedings of ACMMM*, 2020, pp. 184–192.

[25] Z. Wang, Y. Zou, and Z. Zhang, “Cluster Attention Contrast for Video Anomaly Detection,” in *Proceedings of ACMMM*, 2020, pp. 2463–2471.

[26] P. Wu, J. Liu, and F. Shen, “A Deep One-Class Neural Network for Anomalous Event Detection in Complex Scenes,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2609–2622, 2019.

[27] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, “Cloze Test Helps: Effective Video Anomaly Detection via Learning to Complete Video Events,” in *Proceedings of ACMMM*, 2020, pp. 583–591.

[28] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, “Old is Gold: Redefining the Adversarially Learned One-Class Classifier Training Paradigm,” in *Proceedings of CVPR*, 2020, pp. 14 183–14 193.

[29] B. Antic and B. Ommer, “Video parsing for abnormality detection,” in *Proceedings of ICCV*, 2011, pp. 2415–2422.

[30] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression,” in *Proceedings of CVPR*, 2015, pp. 2909–2917.

[31] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in *Proceedings of CVPR*, 2016, pp. 733–742.

[32] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video,” in *Proceedings of CVPR*, 2019, pp. 7842–7851.

[33] J. Kim and K. Grauman, “Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates,” in *Proceedings of CVPR*, 2009, pp. 2921–2928.

[34] W. Li, V. Mahadevan, and N. Vasconcelos, “Anomaly detection and localization in crowded scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2014.

[35] W. Liu, W. Luo, D. Lian, and S. Gao, “Future Frame Prediction for Anomaly Detection – A New Baseline,” in *Proceedings of CVPR*, 2018, pp. 6536–6545.

[36] C. Lu, J. Shi, and J. Jia, “Abnormal Event Detection at 150 FPS in MATLAB,” in *Proceedings of ICCV*, 2013, pp. 2720–2727.

[37] W. Luo, W. Liu, and S. Gao, “A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework,” in *Proceedings of ICCV*, 2017, pp. 341–349.

[38] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, “Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 1975–1981.

[39] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in *Proceedings of CVPR*, 2009, pp. 935–942.

[40] B. Ramachandra, M. Jones, and R. Vatsavai, “Learning a distance function with a Siamese network to localize anomalies in videos,” in *Proceedings of WACV*, 2020, pp. 2598–2607.

[41] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, “Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection,” in *Proceedings of WACV*, 2018, pp. 1689–1698.

[42] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, “Abnormal Event Detection in Videos using Generative Adversarial Nets,” in *Proceedings of ICIP*, 2017, pp. 1577–1581.

[43] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, “Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[44] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, “Integrating prediction and reconstruction for anomaly detection,” *Pattern Recognition Letters*, vol. 129, pp. 123–130, 2020.

[45] D. Xu, Y. Yan, E. Ricci, and N. Sebe, “Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion,” *Computer Vision and Image Understanding*, vol. 156, pp. 117–127, 2017.

[46] B. Zhao, L. Fei-Fei, and E. P. Xing, “Online Detection of Unusual Events in Videos via Dynamic Sparse Coding,” in *Proceedings of CVPR*, 2011, pp. 3313–3320.

[47] X. Zhang, S. Yang, J. Zhang, and W. Zhang, “Video Anomaly Detection and Localization using Motion-field Shape Description and Homogeneity Testing,” *Pattern Recognition*, p. 107394, 2020.

[48] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, “Attention guided anomaly localization in images,” in *Proceedings of ECCV*, 2020, pp. 485–503.

[49] Y. Cong, J. Yuan, and J. Liu, “Sparse reconstruction cost for abnormal event detection,” in *Proceedings of CVPR*, 2011, pp. 3449–3456.

[50] J. K. Dutta and B. Banerjee, “Online Detection of Abnormal Events Using Incremental Coding Length,” in *Proceedings of AAAI*, 2015, pp. 3755–3761.

[51] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund, “Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection,” in *Proceedings of BMVC*, 2015, pp. 28.1–28.13.

[52] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings,” in *Proceedings of CVPR*, 2020, pp. 4183–4192.

[53] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, “Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes,” *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[54] V. Saligrama and Z. Chen, “Video anomaly detection based on local statistical aggregates,” in *Proceedings of CVPR*, 2012, pp. 2112–2119.

[55] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, “Deep Appearance Features for Abnormal Behavior Detection in Video,” in *Proceedings of ICIAP*, vol. 10485, 2017, pp. 779–789.

[56] Q. Sun, H. Liu, and T. Harada, “Online growing neural gas for anomaly detection in changing surveillance scenes,” *Pattern Recognition*, vol. 64, no. C, pp. 187–201, Apr. 2017.

[57] H. T. Tran and D. Hogg, “Anomaly Detection using a Convolutional Winner-Take-All Autoencoder,” in *Proceedings of BMVC*, 2017.

[58] A. Del Giorno, J. Bagnell, and M. Hebert, “A Discriminative Framework for Anomaly Detection in Large Videos,” in *Proceedings of ECCV*, 2016, pp. 334–349.

[59] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, “Unmasking the abnormal events in video,” in *Proceedings of ICCV*, 2017, pp. 2895–2903.

[60] Y. Liu, C.-L. Li, and B. Póczos, “Classifier Two-Sample Test for Video Anomaly Detections,” in *Proceedings of BMVC*, 2018.

[61] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[62] Y. Feng, Y. Yuan, and X. Lu, “Learning deep event models for crowd anomaly detection,” *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[63] R. Hinami, T. Mei, and S. Satoh, “Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge,” in *Proceedings of ICCV*, 2017, pp. 3639–3647.

[64] B. Saleh, A. Farhadi, and A. Elgammal, “Object-Centric Anomaly Detection by Attribute-Based Reasoning,” in *Proceedings of CVPR*, 2013, pp. 787–794.

[65] S. Wu, B. E. Moore, and M. Shah, “Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 2054–2060.

[66] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *Proceedings of CVPR*, 2018, pp. 7132–7141.

[67] C. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-Supervised Learning for Anomaly Detection and Localization,” in *Proceedings of CVPR*, 2021, pp. 9664–9674.

[68] V. Zavrtanik, M. Kristan, and D. Skocaj, “DRAEM – A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection,” in *Proceedings of ICCV*, 2021, pp. 8330–8339.

[69] M. Astrid, M. Z. Zaheer, and S.-I. Lee, “Synthetic Temporal Anomaly Guided End-to-End Video Anomaly Detection,” in *Proceedings of ICCVW*, 2021, pp. 207–214.

[70] D. Zimmerer, S. Kohl, J. Petersen, F. Isensee, and K. Maier-Hein, “Context-encoding Variational Autoencoder for Unsupervised Anomaly Detection,” in *Proceedings of MIDL*, 2019.

[71] M. Haselmann, D. P. Gruber, and P. Tabatabai, “Anomaly detection using deep learning based image completion,” *Proceedings of ICMLA*, pp. 1237–1242, 2018.

[72] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, “Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning,” in *Proceedings of AAAI*, 2020, pp. 11 701–11 708.

[73] M. Sabokrou, M. PourReza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, “AVID: Adversarial Visual Irregularity Detection,” in *Proceedings of ACCV*, 2018, pp. 488–505.

[74] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[75] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proceedings of NIPS*, 2012, pp. 1106–1114.

[76] X. Guo, Z. Jin, C. Chen, H. Nie, J. Huang, D. Cai, X. He, and X. Hua, "Discriminative-Generative Dual Memory Video Anomaly Detection," *arXiv preprint arXiv:2104.14430*, 2021.

[77] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Proceedings of ECCV*, 2014, pp. 818–833.

[78] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules," in *Proceedings of NIPS*, 2017, pp. 3859–3869.

[79] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of ICML*, 2010, pp. 807–814.

[80] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization," in *Proceedings of ICCV*, 2017, pp. 618–626.

[81] W. Sultani, C. Chen, and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," in *Proceedings of CVPR*, 2018, pp. 6479–6488.

[82] S. Lee, H. G. Kim, and Y. M. Ro, "STAN: Spatio-temporal adversarial networks for abnormal event detection," in *Proceedings of ICASSP*, 2018, pp. 1323–1327.

[83] M. Astrid, M. Z. Zaheer, J.-Y. Lee, and S.-I. Lee, "Learning not to reconstruct anomalies," in *Proceedings of BMVC*, 2021.

Paper D

Self-Supervised Predictive Convolutional Transformer Block for Anomaly Detection

Neelu Madan, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B. Moeslund, Mubarak Shah

The paper has been submitted to
IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2023

Abstract

Anomaly detection has recently gained increasing attention in the field of computer vision, likely due to its broad set of applications ranging from product fault detection on industrial production lines and impending event detection in video surveillance to finding lesions in medical scans. Regardless of the domain, anomaly detection is typically framed as a one-class classification task, where the learning is conducted on normal examples only. An entire family of successful anomaly detection methods is based on learning to reconstruct masked normal inputs (e.g. patches, future frames, etc.) and exerting the magnitude of the reconstruction error as an indicator for the abnormality level. Unlike other reconstruction-based methods, we present a novel self-supervised masked convolutional transformer block (SSMCTB) that comprises the reconstruction-based functionality at a core architectural level. The proposed self-supervised block is extremely flexible, enabling information masking at any layer of a neural network and being compatible with a wide range of neural architectures. In this work, we extend our previous self-supervised predictive convolutional attentive block (SSPCAB) with a 3D masked convolutional layer, a transformer for channel-wise attention, as well as a novel self-supervised objective based on Huber loss. Furthermore, we show that our block is applicable to a wider variety of tasks, adding anomaly detection in medical images and thermal videos to the previously considered tasks based on RGB images and surveillance videos. We exhibit the generality and flexibility of SSMCTB by integrating it into multiple state-of-the-art neural models for anomaly detection, bringing forth empirical results that confirm considerable performance improvements on five benchmarks: MVTec AD, BRATS, Avenue, ShanghaiTech, and Thermal Rare Event. We release our code and data as open source at: https://github.com/ristea/ssmctb.

1 Introduction

The applications of vision-based anomaly detection are very diverse, ranging from industrial settings, where the need is to detect faulty objects in the production line [1, 2], to video surveillance, where the need is to detect abnormal behavior [3] such as people fighting or shoplifting, and even medical imaging, where the need is to detect abnormal tissue [4] such as malignant lesions. One of the major challenges of the anomaly detection task is that the definition of what represents an anomaly implies a high dependence on context. For instance, a car driven in a pedestrian area is labeled as anomalous, whereas the same action can be considered normal in a different context, *e.g*. when the car is driven on the road. Due to the reliance on context and the sheer diversity of possible anomalies, it is often very difficult to gather abnormal examples for training. As a result, anomaly detection is commonly devised as a one-class classification task, where the generic approach implicitly or explicitly learns the distribution of the normal training data. During inference, examples that do not belong to the nor-

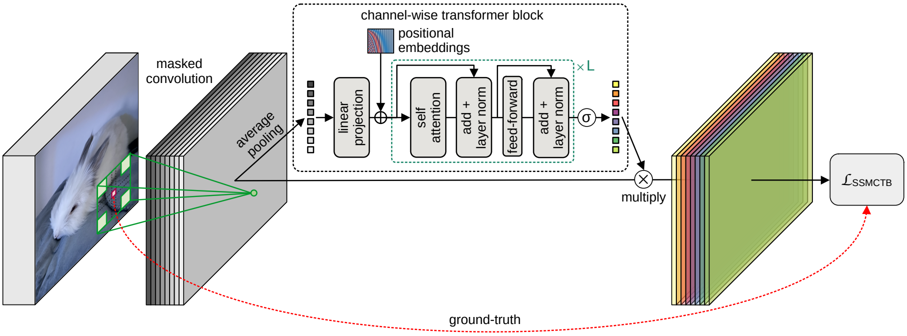


Fig. D.1: An overview of our self-supervised masked convolutional transformer block (SSMCTB). At every location where the masked filters are applied, the proposed block has to rely on the visible regions (sub-kernels) to reconstruct the masked region (center area). A transformer module performs channel-wise self-attention to selectively promote or suppress reconstruction maps via a set of weights returned by a sigmoid (σ) layer. The block is self-supervised via the Huber loss ($\mathcal{L}_{\text{SSMCTB}}$) [49] between masked and returned activation maps. Best viewed in color.

mal training data distribution are labeled as abnormal. There are several categories of methods that are guided by this generic approach, such as dictionary-learning methods [5–10], change-detection frameworks [11–14], distance-based models [15–27], probabilistic frameworks [28–37], and reconstruction-based models [3, 38–48].

Our approach belongs to the category of reconstruction methods, which have recently become a prominent choice in anomaly detection [38, 39, 41, 43, 44, 46–48]. Reconstruction-based models implicitly learn the normal data distribution by minimizing the reconstruction error of the normal instances at training time. These models are based on the assumption that the learned latent manifold does not offer the means to reconstruct the abnormal samples robustly, due to the unavailability of such samples at training time. Hence, the reconstruction error is directly employed as the anomaly score. A particular subcategory of reconstruction-based models relies on learning to predict masked inputs [41, 42, 50, 51] as a self-supervised pretext task. In this case, the reconstruction error with respect to the masked information is used to assess the abnormality level of an input instance. Depending on the input type (image or video), methods in this subcategory mask various parts of the input, *e.g.* superpixels in images [41], future frames in video [42], or middle bounding boxes in object-centric temporal sequences [50, 51], and employ the whole model to reconstruct the masked input. We, on the other hand, propose to encapsulate the functionality of reconstructing the masked information into a novel neural block. There are two major benefits when wrapping the reconstruction task as a low-level architectural component: (i) it enables introducing the reconstruction of masked information as a self-supervised task at any layer of a neural network (not only at the input), and (ii) it eases integrating the self-supervised reconstruction task into a broad variety of neural architectures, regardless of whether the respective models are reconstruction-based or not. Due to its advantages, our block is very flexible and generic.

Our self-supervised reconstruction block consists of a dilated masked convolution followed by a channel-wise transformer module. The center area of our convolutional kernel is masked, hence hiding the center of the receptive field at every location where the filters are applied. In other words, each component of the input tensor is certainly masked at some point during the convolution operation, which means that the entire input tensor ends up being masked. Next, the convolutional activation maps are transformed into tokens using an average pooling layer. Then, the resulting tokens are passed through a transformer module [52, 53] that performs channel-wise self-attention. The proposed block is equipped with a transformer module to avoid the direct reconstruction of the masked area through linearly interpolating the visible regions of the convolutional kernels. The final activation maps are multiplied with the resulting attention tokens. Our block is designed in such a way that the output tensor has the same dimensions as the input tensor, which allows us to easily introduce a loss within our block to minimize the reconstruction error between the output tensor and the masked input tensor. By integrating this loss, our block becomes a self-contained trainable component that learns to predict the masked information via self-supervision. As such, we coin the term *self-supervised masked convolutional transformer block* (SSMCTB) to designate our novel neural component for anomaly detection. As shown in Figure D.1, SSMCTB learns to reconstruct the masked region based on the available context (visible regions of the receptive field), for each location where the dilated kernels are applied. Notably, we can graciously control the level (from local to global) of the contextual information by choosing the appropriate dilation rate for the masked kernels.

SSMCTB is an extension of the self-supervised predictive convolutional attentive block (SSPCAB) introduced in our recent CVPR 2022 paper [54]. In the current work, we modify SSPCAB in three different ways: (i) we replace the standard channel attention module in the original SSPCAB [54] with a multi-head self-attention module [52, 53] to increase the modeling capacity, (ii) we extend the masked convolution operation with 3D convolutional filters, enabling the integration of SSMCTB into networks based on 3D convolutional layers, and (iii) we replace the mean squared error (MSE) loss with the Huber loss [49], since the latter loss is less sensitive to outliers than the former loss. Aside from these architectural changes, we demonstrate the applicability of our block to more domains, adding anomaly detection in medical images and thermal videos to the previously considered tasks based on RGB images and surveillance videos. Moreover, we conduct a more extensive ablation study, thus providing a more comprehensive set of results. We also show that our module is suitable for both convolutional and transformer-based architectures.

We introduce SSMCTB into multiple state-of-the-art neural models [42, 44, 55–60] for anomaly detection and conduct experiments on five benchmarks: MVTec AD [1], BRATS [61], Avenue [9], ShanghaiTech [3], and Thermal Rare Event. The Thermal Rare Event data set is a novel benchmark for anomaly detection, which we constructed by manually labeling abnormal events from the Seasons in Drift data set [62]. The chosen benchmarks belong to various domains, ranging from industrial and medical images to RGB and thermal videos. This is to show that SSMCTB is applicable to multiple domains. When adding SSMCTB to the state-of-the-art models, our experiments show evidence of consistent improvements across all models and tasks, indicating that our block is generic and easily adaptable. When compared to SSPCAB, we observe performance gains in the majority of cases, showing that the multi-head self-attention and the Huber loss are beneficial in detriment of the standard channel attention [63] and the MSE loss, respectively.

In summary, with respect to our previous work [54], our current contribution is sixfold:

- We extend the 2D masked convolution to a 3D masked convolution that considers a 3D context, and we integrate the new 3D SSMCTB into two 3D networks for anomaly detection [55, 56].
- We replace the Squeeze-and-Excitation module [63] of SSPCAB with a transformer module that performs channel-wise attention.
- We substitute the MSE loss with the Huber loss, improving the sensitivity to outliers during self-supervised learning.
- We conduct a more comprehensive set of experiments, including a new method and two new benchmarks from previously missing domains (medical images, thermal videos).
- We provide an extensive ablation study, including different variations of the proposed self-supervised block.
- We annotate a subset (one week of video) of the Seasons in Drift [62] data set with anomaly labels, obtaining a new benchmark for anomaly detection in thermal videos.

2 Related Work

2.1 Transformers

Vaswani *et al.* [53] introduced the self-attention mechanism, sparking the research of neural architectures relying solely on attention, including research on vision transformers [52, 64–74]. These models are now embraced at a fast pace in the field of computer vision, certainly due to the imposing performance levels across a broad variety of tasks, ranging from object recognition [52, 69, 70] and object detection [64, 73, 74] to image generation [68, 71, 72] and anomaly detection [75–78]. Unlike approaches using only transformer-based attention [52, 64–70, 73, 74, 79], we propose a novel and flexible block that employs transformer-based attention along with masked convolution, which can be integrated into multiple architectures that are not necessarily transformer-based. To endorse this statement, we introduce SSMCTB into a variety of models and conduct a series of experiments showing that our block can bring significant performance gains. Another difference from vision transformers is that our block performs channel-wise self-attention, while conventional vision transformers perform spatial attention [52]. We conduct an ablation study to compare channel and spatial attention inside SSMCTB, showing that channel attention provides superior performance and faster processing.

2.2 Self-Supervision via Information Masking

The reconstruction of masked information has recently become an attractive area of interest [60, 80–83]. Models based on information masking are usually pre-trained on a self-supervised reconstruction task, being later employed for downstream visual tasks such as object detection and image segmentation. For instance, He *et al.* [60] proposed to reconstruct masked (erased) patches as a self-supervised pretext task for pre-training auto-encoders, subsequently using them for mainstream tasks, including object detection and object recognition. They reported optimal results when a majority (75%) of the patches is masked. Masked auto-encoders are directly applicable to anomaly detection. However, we show that SSMCTB can boost the performance of masked auto-encoders, suggesting that it can leverage information masking in a distinct way. Wei *et al.* [81] aimed at pre-training video models, proposing to mask spatio-temporal cubes from a video and predict the features of the masked regions. Chang *et al.* [82] introduced a bidirectional decoder that learns to predict masked tokens by attending them from all directions. The proposed method provides an efficient substitute for generative transformers. Yu *et al.* [83] used a masked point modeling task for pre-training a point cloud transformer. They showed that the representation learned by the model transfers well to new (downstream) tasks and domains. Distinct from such methods, we integrate information masking at a core operational level inside neural networks via our masked convolutional layer. We self-supervise our block (which incorporates masked convolution) through a reconstruction loss and show that modeling the context towards reconstructing the masked information results in an effective discriminative manifold for anomaly detection.

We underline that some recent approaches [42, 50, 84] utilize masking as a surrogate task for anomaly detection. We discuss these methods and explain how our approach is different in a separate subsection below.

2.3 Anomaly Detection

Anomaly detection frameworks are usually trained in a one-class setting, where only normal data is available at training time, whereas both normal and abnormal examples are present at test time. The anomaly detection methods operating in this setting can be classified into different categories, which are briefly presented below. Dictionary learning methods [5–10] construct a dictionary of atoms from normal instances, labeling examples that are not represented in the dictionary as abnormal.

Change detection frameworks [11–14] are applied directly on test videos, measuring the degree of change between current and preceding video frames to detect anomalies. Probabilistic models [28–37] learn the probability density function of the normal data, flagging examples outside the distribution as abnormal. Distance-based approaches [15–27, 85] learn a distance function between samples, such that the distance between normal instances is lower than the distance between normal and abnormal instances. Reconstruction-based methods [3, 38–48, 56, 86] learn to reconstruct normal examples, detecting anomalies based on the magnitude of the reconstruction error, as anomalies tend to have larger errors than normal instances.

Reconstruction-based methods. Since our block belongs to the category of reconstruction-based models, we discuss this category in more detail next. Reconstruction-based models are often chosen for both image and video anomaly detection [44, 50, 56, 59]. These approaches typically employ auto-encoders and generative adversarial networks (GANs) to learn a powerful latent manifold representing the normal data distribution. For the video domain, some anomaly detection approaches [17, 42, 59] incorporate additional cues by reconstructing the optical flow to capture motion information, enabling the detection of motion-based anomalies such as running and jumping. Doshi *et al.* [87] proposed a continual learning setup, which could be easily extended for future normal and abnormal patterns.

As the amount of normal training data is generally high, latent manifolds show a tendency to generalize too well, being capable of reconstructing abnormal instances with low error. In the context of anomaly detection, generalizing to out-of-distribution samples, *e.g.* anomalies, is not desired, although this would be mostly desirable in other application domains. To mitigate this issue, researchers employed various techniques, such as adding memory modules [39, 44, 59] or pseudo-anomalies during training [58, 84]. Memory-based auto-encoders [39, 59] generally employ an additional module to memorize the normal patterns observed in the training data. Consequently, memory modules increase the computational complexity of the model, and the faithful reconstruction of normal samples highly relies on the size of the memory module. Georgescu *et al.* [58] proposed to optimize the model on pseudo-anomalies with gradient ascent, while still using gradient descent to learn the normal data distribution. This results in a powerful discriminative subspace for the robust detection of the abnormal samples. The pseudo-abnormal instances are samples collected from different contexts, such as flowers, animals, cartoons, and textures, unrelated to the object distribution (comprising humans, cars, bicycles, etc.) observed in typical urban surveillance scenes. Similarly, Astrid *et al.* [84] generated pseudo-anomalies by skipping a few frames from the video and training an auto-encoder by maximizing the loss for pseudo-anomalies and minimizing it for normal samples. Introducing pseudo-anomalies increases the training time and may sometimes cause instability if the balance between gradient descent on normal data and gradient ascent on pseudo-abnormal data is not tuned. Different from related reconstruction-based methods, we increase the difficulty of the reconstruction task by masking information wherever SSMCTB is introduced into a neural model, thus making it harder for the model to generalize to abnormal data. As shown by our experimental results, our block adds a marginal computational overhead.

Masking for Anomaly Detection. Some approaches [38, 42, 50, 75, 78, 88, 89] are already using the prediction of masked inputs as a surrogate task for anomaly detection. These models form a distinctive subcategory of reconstruction-based methods. Liu *et al.* [42] proposed a GAN for predicting a future frame based on a few past frames, where anomalies are classified according to the prediction error. Another GAN-based approach [90] performs joint detection and localization of anomalies via inpainting. The generator of this method learns to inpaint a patch from the input image, while the discriminator learns to identify if the inpainted patch is normal or abnormal. Interestingly, the inpainting task has also been studied in conjunction with vision transformers [78].

Generalizing over the method of Liu *et al.* [42], Yu *et al.* [89] employed the Cloze task [91], which is about learning to complete the video when certain frames are removed. Georgescu *et al.* [50] proposed the masking of the middle box of each temporal cube centered on an object. Anomalies are detected based on the assumption that motion reconstruction for an abnormal object is more difficult than for the normal ones. Fei *et al.* [38] proposed the Attribute Restoration Network (ARNet), where attributes such as color and orientation of the input are removed, and the network learns to restore those attributes. The idea is based on the assumption that the anomalous data can be distinguished based on the restoration error. Haselmann *et al.* [88] introduced an approach for surface anomaly detection by erasing a rectangular box from the center of the image and using the interpolation error for the classification of samples into normal or abnormal. Inspired by the success of masked auto-encoders [60], Jiang *et al.* [75] proposed a masked Swin Transformer [92] that is trained to inpaint masked regions. To cope with the lack of abnormal samples during training, the authors used simulated anomalies.

Unlike other models based on information masking, we propose a novel approach that incorporates the reconstruction-based functionality into a single neural block, which can be easily integrated into other state-of-the-art anomaly detection models. Our experimental results confirm that our block is a valuable addition to various models, including both CNNs and transformers, which are applied to anomaly detection in a wide range of domains.

3 Method

3.1 Motivation and Overview

A wide set of computer vision tasks, including anomaly detection [44, 58, 59, 93, 94], are often addressed with convolutional neural networks (CNNs) [95, 96], due to the impressive performance levels reached by these models, sometimes even surpassing human-level accuracy. The defining component of a CNN architecture is the convolutional layer, which typically comprises multiple filters (kernels) that activate on discriminative local patterns captured within the receptive field of the respective filters. Each filter produces an activation map that is further given as input to the next convolutional layer. Since each filter in the subsequent layer processes all activation maps from the previous layer at once, the local features extracted by the previous layer are combined into more complex features. This sequential processing of features over multiple convolutional layers gives rise to a hierarchy of features during the learning process. Earlier convolutional layers activate on low-level features such as corners or edges, and later layers gradually shift to higher-level features such as car wheels or human body parts, as shown by Zeiler *et al.* [97]. Although the learned hierarchy of features is very useful in solving discriminative tasks, CNNs do not have the direct means to model the global arrangement of local features [98], since they do not generalize well to novel viewpoints or affine transformations [99]. The inability of grasping the global arrangement of local features is mainly caused by the fact that convolutional filters operate on a limited (and typically small) receptive field, not making use of the context.

We hereby propose a self-supervised masked convolutional transformer block (SSMCTB), which is aimed at learning to reconstruct masked information based on contextual information. To accurately solve the reconstruction of its masked input, the proposed block is required to employ the context and learn the global structure of the local patterns. Hence, it inherently learns to cope with the problem stated by Sabour *et al.* [98], specifically the fact that CNNs lack the proper comprehension of the global arrangement of local features. To embed this learning capability into our block, we structure SSMCTB as a convolutional layer with dilated masked kernels, followed by a transformer module that performs channel attention. We attach a self-supervised loss function to our block in order to minimize the reconstruction error between the masked input and the predicted output.

We emphasize that SSMCTB is quite flexible, since it can be inserted at any level of almost any CNN or transformer model, generating powerful features that offer the capability of reconstructing masked information based on context. While the ability of learning and harnessing the global arrangement of local patterns is potentially useful in solving a broader set of computer vision tasks, we conjecture that anomaly detection is a natural and immediate application domain for SSMCTB, hence focusing our work in this direction. Indeed, since anomaly detection models are typically trained on normal data only, integrating SSMCTB into a neural model will lead to the learning of features that recover only masked normal data. Hence, when an anomalous sample is given as input during inference, SSMCTB is likely less capable of reconstructing the masked information. This empowers the model to directly estimate the abnormality level of a data sample via the reconstruction error given by SSMCTB. Our claims are supported through the comprehensive set of experiments on image and video anomaly detection presented in Section 4.

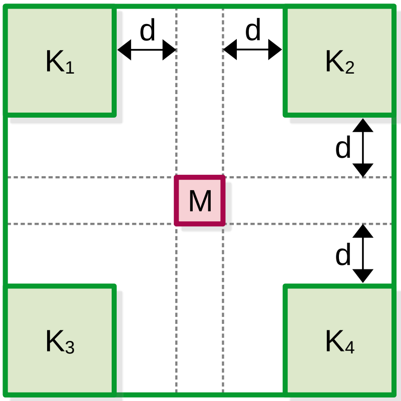


Fig. D.2: Our 2D masked convolutional kernel. The visible area of the receptive field is denoted by the regions $\boldsymbol{K}_i, \forall i \in \{1,2,3,4\}$, while the masked area is denoted by $\boldsymbol{M}$. A dilation factor d controls the local or global nature of the visible information with respect to $\boldsymbol{M}$. Best viewed in color.

3.2 Architecture

Our initial self-supervised block introduced in [54] was formed of a 2D masked convolution and a Squeeze-and-Excitation (SE) module [63]. To broaden the applicability of our block, we now introduce a 3D masked convolutional layer to replace the 2D masked convolution, whenever this is needed. Moreover, we replace the SE attention module with a modern transformer-based attention module [52, 53] to attend to the channels given as output by the masked convolution. We describe the individual components of our block below, while providing a graphical overview of SSMCTB in Figure D.1.

2D Masked Convolution. Figure D.2 shows our 2D masked convolutional kernel, where the corner regions of this kernel (in green color) are the learnable parameters (weights) defining the visible regions of the receptive field. The four learnable sub-kernels are denoted by $\boldsymbol{K}_i \in \mathbb{R}^{k' \times k' \times c}$, $\forall i \in \{1,2,3,4\}$, where the spatial size $k' \in \mathbb{N}^+$ of each sub-kernel is a hyperparameter of our block, while the number of channels $c \in \mathbb{N}^+$ always matches the number of channels of the input tensor. Our masked region $\boldsymbol{M} \in \mathbb{R}^{1 \times 1 \times c}$ (in pink color) is located at the center of the receptive field. Each sub-kernel $\boldsymbol{K}_i$ is located at a configurable distance $d \in \mathbb{N}^+$ (also referred to as *dilation rate*) from the masked region $\boldsymbol{M}$. To keep the number of hyperparameters to a bare minimum, we fix the spatial size of the masked region to 1×1. As a result, the spatial size k of the entire receptive field of our 2D masked convolution is $k = 2k' + 2d + 1$.

Let $\boldsymbol{X} \in \mathbb{R}^{h \times w \times c}$ be the input tensor of the masked convolutional layer, where $c \in \mathbb{N}^+$ denotes the number of channels, and $h, w \in \mathbb{N}^+$ represent the height and width of the input tensor, respectively. When we apply our custom kernel at a given location (a, b) of the input tensor $\boldsymbol{X}$, only the input values that overlap with the sub-kernels $\boldsymbol{K}_i$ are taken into consideration during the masked convolution operation, resulting in a single output value. We underline that our masked convolution is

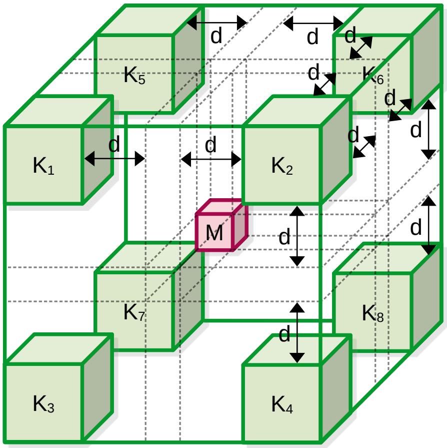


Fig. D.3: Our 3D masked convolutional kernel. The visible area of the receptive field is denoted by the regions $\boldsymbol{K}_i, \forall i \in \{1, 2, ..., 8\}$, while the masked area is denoted by $\boldsymbol{M}$. A dilation factor d controls the local or global nature of the visible information with respect to $\boldsymbol{M}$. Best viewed in color.

equivalent to convolving the input independently with the sub-kernels $\boldsymbol{K}_i$, where each sub-kernel has a different spatial shift with respect to the current location (a, b), and the resulting values are summed up to produce a single output value. The output value at position (a, b) represents the reconstruction for only one value of the tensor $\boldsymbol{M}$ located at the same position (a, b). To reconstruct the entire tensor $\boldsymbol{M}$, our layer requires the application of c masked convolutional filters, each reconstructing the masked value from a distinct channel at position (a, b). Convolving a single masked filter over the entire input generates a complete activation map. Since there are c masked convolutional filters, the output tensor $\boldsymbol{Z}$ is formed of c activation maps. Our aim is to apply the masked convolution such that every element in the input tensor is masked exactly once, *i.e.* we want to mask and predict the reconstruction for every spatial location of the input. As such, we set the stride to 1 and apply a zero-padding of $k' + d$ in each direction. With this configuration in place, the output tensor $\boldsymbol{Z}$ has $h \times w \times c$ components, exactly as the input tensor $\boldsymbol{X}$. To obtain the final values, the output tensor $\boldsymbol{Z}$ is passed through Rectified Linear Units (ReLU) [100]. Finally, we emphasize that k' and d are the only tunable hyperparameters of our masked convolutional layer.

3D Masked Convolution. Considering that anomaly detection is often applied on 3D inputs, *e.g.* video or medical scans, some researchers naturally resort to employing 3D CNNs. To this end, we extend our 2D masked convolution to the 3D domain, broadening the applicability of SSMCTB. We thus reformulate the 2D spatial reconstruction task into a more difficult one, which implies learning a global 3D structure of the discovered local patterns. Let $\boldsymbol{K}_i \in \mathbb{R}^{k' \times k' \times k' \times c}, \forall i \in \{1, 2, ..., 8\}$, be the learnable 3D sub-kernels depicted in Figure D.3, where k' and c are defined above. The masked region $\boldsymbol{M}$ is located in the center of the 3D kernel, equally distant from the sub-kernels $\boldsymbol{K}_i$. The size of the receptive field of our 3D masked convolution is $k \times k \times k$, where

$k = 2k' + 2d + 1$.

To compute the feature response using the 3D masked convolutional layer, the input $\boldsymbol{X} \in \mathbb{R}^{h\times w\times r\times c}$ is convolved with our custom masked kernel, where r represents the depth, and h, w and c are defined as before. The 3D filter is applied analogously to the 2D one, the only difference being that the input data and the kernel itself are 3D. The number of 3D convolutional filters is equal to the number of channels c, such that the spatial dimension of the output tensor $\boldsymbol{Z} \in \mathbb{R}^{h\times w\times r\times c}$ is identical to that of the input $\boldsymbol{X}$. The 3D masked convolution has the same number of configurable hyperparameters, these being k' and d.

Channel-wise transformer block. To better exploit the interdependencies between the different activation maps produced by the masked convolutional layer, we replace the Squeeze-and-Excitation module in SSPCAB [54] with a self-attention transformer-based module. The new attention module is able to capture more complex channel-wise interrelations through its higher modeling capacity, as it learns to assign attention weights to the reconstructed information corresponding to each masked convolutional filter in order to reduce the reconstruction error of SSMCTB.

Let $\boldsymbol{Z} \in \mathbb{R}^{h\times w\times c}$ be the output tensor of a 2D masked convolutional layer with c filters. First, we apply a spatial average pooling, obtaining $\hat{\boldsymbol{Z}} \in \mathbb{R}^{h'\times w'\times c}$, where $h' \leq h$ and $w' \leq w$. The average pooling layer is followed by a reshape operation, obtaining a matrix $\boldsymbol{A} \in \mathbb{R}^{c\times n}$, which contains a vector of $n = h' \cdot w'$ components on each row to represent each masked filter. Next, $\boldsymbol{A}$ is fed into a linear projection layer to obtain the tokens $\boldsymbol{T} \in \mathbb{R}^{c\times d_t}$, which are further summed up with the positional embeddings to obtain the final tokens $\boldsymbol{T}^* \in \mathbb{R}^{c\times d_t}$.

Let f be a multi-head attention layer with $H \in \mathbb{N}^+$ heads, g a multi-layer perceptron, *norm* a normalization layer, and $\boldsymbol{P}, \boldsymbol{R} \in \mathbb{R}^{c\times d_t}$ some auxiliary tensors. The operations performed inside the transformer are formally described as follows:

$$\boldsymbol{P} = f(norm(\boldsymbol{R})) + \boldsymbol{R}, \tag{D.1}$$

$$\boldsymbol{R} = g(norm(\boldsymbol{P})) + \boldsymbol{P}. \tag{D.2}$$

As illustrated in Figure D.1, the whole process described in Eq. (D.1) and Eq. (D.2) is repeated L times, where $L \in \mathbb{N}^+$ represents the number of transformer blocks inside the transformer module. For the first transformer block, $\boldsymbol{R}$ is initialized with $\boldsymbol{T}^*$. In Eq. (D.1), the sequence of c tokens $\boldsymbol{R}$ is normalized, fed into the multi-head attention layer and added to itself, obtaining $\boldsymbol{P}$. Further, $\boldsymbol{P}$ is normalized, fed into a multi-layer perceptron and also added to itself, according to Eq. (D.2).

The transformer is aimed at capturing the interaction among all c tokens by encoding each token in terms of the channel-wise contextual information. This is achieved via the multi-head attention layer f. Each head $j \in \{1, 2, ..., H\}$ comprises three learnable weight matrices denoted as $\boldsymbol{W}^{Q_j} \in \mathbb{R}^{d_t\times d_q}$, $\boldsymbol{W}^{K_j} \in \mathbb{R}^{d_t\times d_k}$ and $\boldsymbol{W}^{V_j} \in \mathbb{R}^{d_t\times d_v}$, where $d_q = d_k$. The weight matrices are multiplied with the input tokens $\boldsymbol{R}$, producing the queries $\boldsymbol{Q}_j$, keys $\boldsymbol{K}_j$ and values $\boldsymbol{V}_j$. In other words, the input sequence $\boldsymbol{R}$ is projected onto these weight matrices to get $\boldsymbol{Q}_j = \boldsymbol{R} \cdot \boldsymbol{W}^{Q_j}$, $\boldsymbol{K}_j = \boldsymbol{R} \cdot \boldsymbol{W}^{K_j}$ and

$\boldsymbol{V}_j = \boldsymbol{R} \cdot \boldsymbol{W}^{V_j}$, respectively. The output $\boldsymbol{Y}_j \in \mathbb{R}^{c \times d_v}$ of each self-attention head is given by:

$$\boldsymbol{Y}_j = softmax\left(\frac{\boldsymbol{Q}_j \cdot \boldsymbol{K}_j^\top}{\sqrt{d_q}}\right) \cdot \boldsymbol{V}_j, \tag{D.3}$$

where $\boldsymbol{K}_j^\top$ is the transpose of $\boldsymbol{K}_j$. The outputs returned by the self-attention heads are simply summed into $\boldsymbol{Y}$, *i.e.*:

$$\boldsymbol{Y} = \sum_{j=1}^{H} \boldsymbol{Y}_j. \tag{D.4}$$

We can now rewrite Eq. (D.1) as follows:

$$\boldsymbol{P} = \boldsymbol{Y} + \boldsymbol{R}. \tag{D.5}$$

The output sequence $\boldsymbol{R}$ returned by the final transformer block is averaged along the token dimension, obtaining $\hat{\boldsymbol{R}} \in \mathbb{R}^{c \times 1}$, then fed into a sigmoid layer to generate the final attention weight assigned to each channel. Finally, the resulting attention weights are applied to the tensor $\boldsymbol{Z}$, obtaining the reconstructed output denoted by $\hat{\boldsymbol{X}} \in \mathbb{R}^{h \times w \times c}$, as follows:

$$\hat{\boldsymbol{X}} = \boldsymbol{Z} \otimes \sigma(\hat{\boldsymbol{R}}), \tag{D.6}$$

where $\otimes$ denotes the element-wise multiplication, and σ denotes the sigmoid layer. The entire processing performed by the transformer module is analogously applied when the preceding layer is a 3D masked convolution.

3.3 Self-Supervised Reconstruction Loss

We devise an integrated reconstruction loss to train the proposed SSMCTB in a self-supervised manner. To better cope with outlier values and reduce the sensitivity of the model to outliers, we define the self-supervised objective as the Huber loss between the reconstructed output $\hat{\boldsymbol{X}}$ and the input $\boldsymbol{X}$, replacing the mean squared error (MSE) used by SSPCAB. The self-supervised objective enables our model to learn reconstructing the masked information at every location where the masked filters are applied. Let G denote the SSMCTB function. With this notation, the self-supervised reconstruction loss of our block can be computed as follows:

$$\begin{aligned}\mathcal{L}_{\text{SSMCTB}}(G,\boldsymbol{X}) &= \begin{cases} \frac{1}{2}\cdot(G(\boldsymbol{X})-\boldsymbol{X})^2, & \text{if } |G(\boldsymbol{X})-\boldsymbol{X}|<\delta \\ \delta\cdot\left(|G(\boldsymbol{X})-\boldsymbol{X}|-\frac{\delta}{2}\right), & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{1}{2}\cdot\left(\hat{\boldsymbol{X}}-\boldsymbol{X}\right)^2, & \text{if } |\hat{\boldsymbol{X}}-\boldsymbol{X}|<\delta \\ \delta\cdot\left(|\hat{\boldsymbol{X}}-\boldsymbol{X}|-\frac{\delta}{2}\right), & \text{otherwise} \end{cases},\end{aligned} \tag{D.7}$$

where $\delta \in \mathbb{R}^+$ is a hyperparameter representing the error threshold that determines when to switch from the squared loss (applied for errors below δ) to the absolute loss (applied for errors higher than or equal to δ).

When integrating SSMCTB into some neural network F, we can simply add our loss $\mathcal{L}_{\text{SSMCTB}}$ to the loss function $\mathcal{L}_F$ of the respective neural model, resulting in a new loss function comprising both terms. Formally, the overall loss can be computed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_F + \lambda \cdot \mathcal{L}_{\text{SSMCTB}}, \tag{D.8}$$

where $\lambda \in \mathbb{R}^+$ is a hyperparameter deciding the importance of $\mathcal{L}_{\text{SSMCTB}}$ with respect to $\mathcal{L}_F$. Naturally, the hyperparameter λ can vary across neural models or visual tasks.

4 Experiments and Results

4.1 Data Sets

We carry out experiments on five benchmarks from various domains, considering the most popular data set choices, *e.g.* MVTec AD [1], BRATS [61], CUHK Avenue [9], ShanghaiTech [3], whenever such an option is available for a certain domain. For the thermal video domain, we build our own data set.

MVTec AD. MVTec AD [1] has become a standard data set for benchmarking anomaly detection methods applied in inspecting industrial defects. The data set contains over 5,000 images distributed over 15 different categories of textures (10) and objects (5). It comprises 3,629 defect-free training samples, as well as 1,725 test images with and without defects.

BRATS. BRATS [61] is a multimodal magnetic resonance imaging (MRI) data set for brain tumor segmentation. It is an intrinsically heterogeneous data set that contains brain tumors of different shape, appearance and histology. The data set comprises manually annotated MRI scans acquired by 19 institutions employing different clinical protocols. To evaluate anomaly detection models, we introduce a novel split of the data set, such that all training images are lesion-free, *i.e.* all images with lesions are kept for testing. The training set includes 11,280 slices (125 scans), which leaves 27,745 slices (180 scans) for the test set.

Avenue. CUHK Avenue [9] is one of the most widely-used data sets for video anomaly detection. It contains 16 videos for training and 21 videos for testing. The training videos comprise only normal events, whereas the test videos contain both normal and abnormal events. The data set contains videos from a single surveillance camera. Avenue contains people-related anomalies such as running, walking in the wrong direction, jumping, dancing, loitering and throwing objects.

ShanghaiTech. ShanghaiTech [3] is one of the largest benchmarks for video anomaly detection, comprising 330 training and 107 test videos. As in CUHK Avenue, abnormal instances appear only at test time. The data set includes videos from multiple scenes. Examples of anomalies are related to people, *e.g.* fighting, jumping and stealing, as well as vehicles, *e.g.* bikes and cars in pedestrian (forbidden) zones.

Thermal Rare Event. To construct the Thermal Rare Event data set, we sampled one week of videos (330 clips) from the Seasons in Drift (SiD) data set [62]. The

Table D.1: Rare events in our thermal anomaly detection data set along with the frequency of each event type.

Rare Event Type	Frequency
Activities in restricted zones	6
Jumping	4
Reverse driving	2
Unexpected activities	2
Unexpected interactions	14
Unexpected vehicle	1
Total	29

SiD data set [62] is an unlabeled thermal surveillance data set captured from a single view over a period of 8 months. The data set captures activities near a harbor front during day and night. Each clip is about 2 minutes long and contains 120 frames, being sampled at 1 frame per second (FPS). Out of the 330 clips, there are 29 clips containing rare (anomalous) events. We manually annotated these rare events at the frame level. In total, our Thermal Rare Event data set contains 36,120 frames for testing and 3,480 frames for training. The list of rare events in our data set along with their respective frequencies are summarized in Table D.1. Examples of rare events from different categories are: activities in restricted zones (people sitting, standing, and running close to the pier), jumping (person jumping, group jumping), unexpected activities (doing yoga, smoking), unexpected interactions (running with stroller, embarking to a boat, debarking from a boat, chasing, dancing), unexpected vehicles (different types of trucks). We release the Thermal Rare Event data set along with our code at: https://github.com/ristea/ssmctb/.

4.2 Evaluation Measures

Image Anomaly Detection. Following Bergmann *et al.* [1], we carry out the evaluation on MVTec AD and BRATS considering the area under the receiver operating characteristics curve (AUROC) and the average precision (AP). To generate the ROC curve, the true positive rate (TPR) is plotted against the false positive rate (FPR). We evaluate both detection and localization performance levels of anomaly detection methods. In anomaly detection, TPR is the proportion of images correctly classified as abnormal, and FPR is the proportion of normal images wrongly classified as abnormal. For the localization task, TPR denotes the proportion of correctly classified abnormal pixels, while FPR represents the proportion of normal pixels incorrectly classified as abnormal. For the localization task, we obtain anomaly segments by applying a threshold to produce a binary decision for each pixel, as described in [1]. The localization AP is obtained by taking the mean at different threshold levels.

Video Anomaly Detection. As the majority of previous works [101], we evaluate the detection performance of video anomaly detection methods using the frame-level area under the curve (AUC). To compute the AUC measure, a video frame is marked as abnormal if at least one pixel is abnormal. Inspired by Georgescu *et al.* [58], we employ both micro AUC and macro AUC. The micro AUC is computed by first concatenating all frames in all videos into a single video, while the macro AUC represents the average of the AUC scores which are independently computed for each single video in the test set. To evaluate the localization performance, we report the region-based detection criterion (RBDC) and the track-based detection criterion (TBDC) proposed by Ramachandra *et al.* [19]. RBDC considers each detected region, marking it as a *true positive* if the intersection over union (IOU) between the detected and the ground-truth anomalous region is greater than α. TBDC marks each tracked region as a *true positive* if the overlap with the ground-truth anomalous track is greater than β. We set the same values for α and β as previous works [19, 58], *i.e.* $\alpha = 0.1$ and $\beta = 0.1$.

4.3 Implementation Details

We choose eight state-of-the-art approaches [42, 44, 55–60] for image and video anomaly detection to serve as underlying models, on top of which we add SSPCAB [54] and SSMCTB (ours). We alternatively integrate SSPCAB and SSMCTB directly into the official implementations of the chosen baselines, while preserving all hyperparameter values, *e.g.* the number of epochs and the learning rate, as specified in the corresponding papers [42, 44, 55–60]. Even so, we are unable to exactly reproduce the original results for two baselines methods, *i.e.* those of Park *et al.* [44] and Liu *et al.* [42]. However, our reproduced quantitative results are still close to the originally reported results. For a fair comparison, we compare the models based on SSPCAB and SSMCTB with the reproduced baselines. Additionally, when we repurpose the approach of Park *et al.* [44] from the RGB domain to the thermal domain, we modify some hyperparameters, namely the number of epochs and the mini-batch size.

Following Ristea *et al.* [54], we replace the penultimate convolutional layer with SSMCTB in most underlying models. One exception is the architecture of Georgescu *et al.* [50], where SSPCAB and SSMCTB are integrated into the penultimate convolutional layer of the decoder instead of the final classification network. Another exception is the masked auto-enconder [60] based on the ViT backbone, where we place SSPCAB and SSMCTB before the first transformer block.

In our previous work [54], we conducted a set of preliminary experiments to find an optimal value for the hyperparameter λ representing the contribution of our self-supervised loss to the total loss defined in Eq. (D.7), taking values from 0.1 to 1 at an interval of 0.1. Following our previous work [54], we keep $\lambda = 0.1$ across all data sets. However, for two baselines [57, 59], we notice that the magnitude of our loss is too high with respect to the original losses of the respective models, dominating the optimization. Following our previous work [54], we decrease λ to 0.001 to reduce the dominant influence of our loss on these two particular models [57, 59].

For the channel-wise transformer, we fix the activation map size after the average pooling layer to 1×1, the token size d_t to 64, the number of heads H to 4, as well

Table D.2: Micro AUC scores (in %) obtained on the Avenue data set with different hyperparameter configurations, varying the kernel size (k'), the dilation rate (d), the loss type, and the attention type, while integrating SSMCTB into the method of Park *et al.* [44]. The top score is highlighted in bold.

Method	$\mathcal{L}_{\text{SSMCTB}}$	d	k'	Attention	Micro AUC
Park *et al.* [44]	-	-	-	-	82.8
+SSMC (no attention)	MAE	0	1	-	83.1
		1	1		83.5
		2	1		84.2
		3	1		84.4
+SSMCTB	MAE	0	1	CA	83.7
		1	1		84.9
		2	1		85.5
		3	1		85.9
	MSE	0	1	CA	84.9
		1	1		85.7
		2	1		85.4
		3	1		86.4
	SSIM	0	1	CA	83.3
		1	1		85.5
		2	1		84.9
		3	1		83.0
	Huber	0	1	CA	84.2
		1	1		**87.0**
		2	1		86.5
		3	1		86.1
	Huber	0	2	CA	84.1
		1	2		84.9
		2	2		84.8
		3	2		86.0
	Huber	0	3	CA	84.5
		1	3		85.0
		2	3		86.4
		3	3		84.3
	Huber	0	1	SA	86.2
		1	1		84.7
		2	1		85.8
		3	1		80.4
	Huber	1	1	CA + SA	86.2

as the number of successive transformer blocks L to 2. We discuss results for other transformer configurations in Section 4.7.

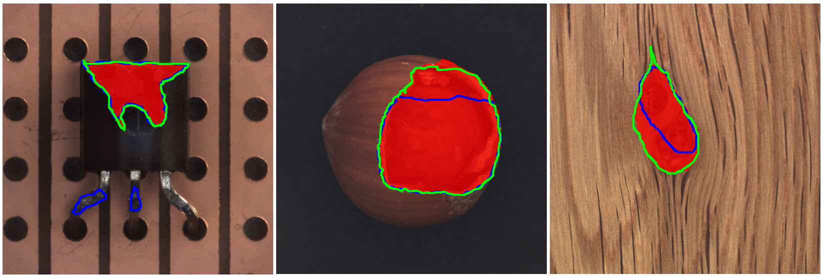


Fig. D.4: Examples of image-level anomaly localization results from MVTec AD given by DRAEM [56], before (blue contour) and after (green contour) integrating SSMCTB. The ground-truth anomalies are shown in red. Best viewed in color.

4.4 Preliminary Results

We conduct a series of preliminary experiments on Avenue to determine the hyperparameters of SSMCTB, namely the dilation rate d and the sub-kernel size k'. We perform experiments with $d \in \{0,1,2,3\}$ and $k' \in \{1,2,3\}$. We also consider alternative attention types, namely channel attention (CA), spatial attention (SA) and both channel and spatial attention (CA+SA). Additionally, we alternate between multiple losses to self-supervise our block, such as the mean absolute error (MAE), the mean squared error (MSE), the Huber loss, and the Structured Similarity Index Measure (SSIM) loss. For the Huber loss, we set the hyperparameter δ to the default value, *i.e.* $\delta = 1$.

We employ the method of Park *et al.* [44] in our preliminary experiments, since this is the most lightweight method among the chosen ones [42, 44, 55–60]. The corresponding micro AUC scores are presented in Table D.2. Except for a single SSMCTB configuration based on spatial attention (SA), all other SSMCTB configurations bring performance improvements over the approach of Park *et al.* [44] (first row). Our first set of preliminary experiments is aimed at evaluating the capacity of the standalone masked convolution. Even without the attention module, our masked convolution brings gains higher than 1% for $d = 2$ and $d = 3$. While adding the attention module is definitely useful, we conclude that it is clearly not the only factor responsible for the reported performance gains. To compare the losses on the one hand, and attention types on the other, we fix $k' = 1$. When alternating between MAE, MSE, SSIM and Huber as our self-supervised loss, we generally observe higher performance with Huber loss. We thus continue the experiments with Huber loss. Regarding the attention type, we note that channel attention (CA) generally leads to better results than spatial attention (SA). Hence, for the remaining experiments, we employ the transformer module based on channel attention. We continue by increasing the size of the sub-kernels, without obtaining further performance gains. We obtain the best micro AUC (86.7%) with $d = 1$ and $k' = 1$, while using channel attention. We make another attempt to further boost the performance by combining the channel and spatial attention (CA+SA), while fixing $d = 1$ and $k' = 1$. This attempt is also unsuccessful. Our final

Table D.3: Detection AUROC and localization AUROC/AP (in %) of two state-of-the-art methods [56, 57] on MVTec AD, before and after alternatively adding SSPCAB and SSMCTB. The best result for each model and each performance measure is highlighted in bold.

	Class	Detection						Localization								
		DRAEM [56]			NSA (logistic) [57]			DRAEM [56]						NSA (logistic) [57]		
		AUROC			AUROC			AUROC			AP			AUROC		
		Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB	Baseline	+SSPCAB	+SSMCTB
Texture	Carpet	97.0	**98.2**	96.8	95.6	**97.5**	96.1	95.5	95.0	**95.8**	53.5	**59.4**	55.2	95.5	**97.5**	95.6
	Grid	99.9	**100**	**100**	99.9	99.9	**100**	**99.7**	99.5	**99.7**	65.7	61.1	**69.7**	**99.2**	**99.2**	**99.2**
	Leather	**100**	**100**	**100**	99.9	99.9	**100**	98.6	**99.5**	97.6	75.3	**76.0**	65.5	99.5	99.5	**99.6**
	Tile	99.6	**100**	**100**	**100**	**100**	**100**	99.2	**99.3**	**99.3**	92.3	95.0	**95.7**	**99.3**	99.2	99.1
	Wood	99.1	99.5	**100**	97.5	97.7	**97.8**	96.4	**96.8**	94.8	**77.7**	77.1	75.6	90.7	90.4	**93.5**
Object	Bottle	99.2	98.4	**99.4**	**97.7**	**97.7**	**97.7**	99.1	98.8	**99.2**	86.5	87.9	**89.9**	98.3	98.3	**98.4**
	Cable	91.8	**96.9**	94.1	94.5	95.6	**96.1**	94.7	**96.0**	95.5	52.4	57.2	**61.6**	96.0	96.6	**97.5**
	Capsule	98.5	**99.3**	97.1	95.2	95.4	**95.5**	**94.3**	93.1	93.4	49.4	50.2	**52.0**	97.6	97.2	**97.9**
	Hazelnut	**100**	**100**	**100**	94.7	94.2	**97.1**	99.7	**99.8**	99.5	**92.9**	92.6	89.1	97.6	**97.9**	**97.9**
	Metal Nut	98.7	**100**	**100**	98.7	99.0	**99.5**	**99.5**	98.9	99.3	96.3	**98.1**	94.7	98.4	**98.6**	98.3
	Pill	98.9	**99.8**	**98.8**	99.2	99.2	**99.5**	**97.6**	97.5	97.4	48.5	**52.4**	46.9	98.5	**98.8**	98.4
	Screw	93.9	97.9	**99.0**	90.2	**91.1**	90.4	97.6	**99.8**	99.5	58.2	**72.0**	70.1	**96.5**	96.2	96.4
	Toothbrush	**100**	**100**	**100**	**100**	**100**	**100**	98.1	98.1	**99.0**	44.7	51.0	**69.0**	94.9	95.3	**95.4**
	Transistor	93.1	92.9	**96.0**	95.1	95.6	**96.2**	**90.9**	87.0	89.1	**50.7**	48.0	45.8	88.0	87.1	**88.3**
	Zipper	**100**	**100**	**100**	99.8	99.8	**99.9**	98.8	**99.0**	**99.0**	**81.5**	77.1	76.5	94.2	94.5	**94.7**
	Overall	98.0	**98.9**	98.7	97.2	97.5	**97.7**	**97.3**	97.2	97.2	68.4	70.3	**70.5**	96.3	96.4	**96.7**

Table D.4: Detection AUROC and localization AUROC/AP (in %) of two state-of-the-art methods [56, 57] on BRATS, before and after alternatively adding SSPCAB and SSMCTB. Additional results obtained by converting DRAEM to use 3D convolutions and integrating the 3D SSMCTB are also reported. The best result for each model and each performance measure is highlighted in bold.

Method	AUROC		Localization AP
	Detection	Localization	
DRAEM [56]	41.06	42.40	45.41
DRAEM + SSPCAB [54]	44.19	46.66	46.89
DRAEM + SSMCTB (Ours)	**50.27**	**53.98**	**50.75**
NSA [57]	53.66	74.90	61.09
NSA + SSPCAB [54]	54.91	75.30	62.37
NSA + SSMCTB (Ours)	**60.09**	**77.09**	**64.55**
3D DRAEM [56]	43.74	44.12	45.97
3D DRAEM + 3D SSMCTB (Ours)	**53.70**	**58.47**	**52.79**

SSMCTB configuration, which we employ across all underlying models and data sets, is based on $d = 1$, $k' = 1$ and channel attention.

We underline that the corresponding hyperparameters for SSPCAB were tuned in a similar manner, in our previous work [54]. Hence, we simply use the already tuned hyperparameters for SSPCAB. Importantly, we underline that our observations above are mostly consistent with those reported in our previous work [54], *i.e.* both SSPCAB and SSMCTB use channel attention, a dilation rate of $d = 1$ and sub-kernels of size $k' = 1$. The only difference is that SSMCTB is based on the Huber loss instead of the MSE loss. We should also emphasize that it is not common for anomaly detection data sets to have validation splits. Since the training set contains normal instances only, keeping a representative training subset (with both normal and abnormal examples) for validation is not possible. This is the reason behind our decision to avoid hyperparameter tuning for each model and data set. We believe that this evaluation procedure is more fair because it avoids overfitting in hyperparameter space.

4.5 Anomaly Detection in Images

Baselines. We introduce SSMCTB into two state-of-the-art baselines for image anomaly detection on MVTec AD, namely a self-supervised model based on natural synthetic anomalies (NSA) [57] and a discriminatively trained reconstruction anomaly embedding model (DRAEM) [56]. Both baselines are very recent, attaining strong results on MVTec AD. The NSA approach of Shülter *et al.* [57] generates synthetic anomalies using Poisson image editing, blending scaled patches of different sizes from separate images. In this way, it generates a wide range of synthetic anomalies that are similar to natural irregularities. DRAEM [56] comprises a reconstructive network and a dis-

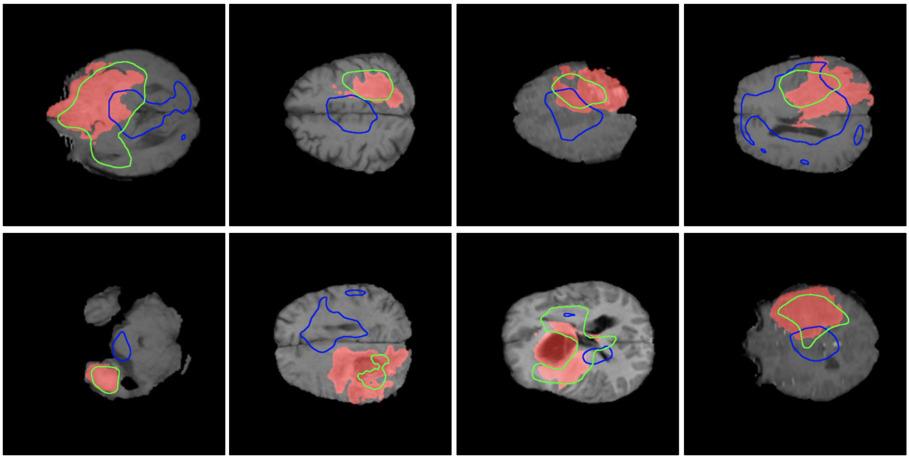


Fig. D.5: Examples of image-level anomaly localization results from BRATS given by DRAEM [56], before (blue contour) and after (green contour) integrating SSMCTB. The ground-truth anomalies are shown in red. Best viewed in color.

criminative network to detect and localize anomalies. The reconstructive network is based on a simple auto-encoder architecture which learns to reconstruct original images from artificially corrupted images. The discriminative network is a U-Net that learns to segment the introduced artifacts (corrupted regions).

Results on MVTec AD. We report the results on MVTec AD in Table D.3. Considering the detection results, we observe that adding SSPCAB and SSMCTB leads to superior results for both DRAEM [56] and NSA [57]. Considering the localization results, the AUROC scores of DRAEM do not show any improvements when adding SSPCAB and SSMCTB. However, the localization AP of DRAEM exhibits gains of around 2% by adding SSPCAB and SSMCTB. In addition, the localization AUROC of NSA grows when SSPCAB and SSMCTB are introduced into the architecture.

In Figure D.4, we present some examples of qualitative results from MVTec AD, obtained by DRAEM [56], before and after adding SSMCTB. In all shown cases, we observe that the anomaly localization results are better aligned with the ground-truth regions when SSMCTB is integrated into DRAEM.

Results on BRATS. In Table D.4, we present the brain lesion detection and localization results obtained by the anomaly detection models [56, 57] on BRATS, before and after adding SSPCAB and SSMCTB, respectively. Remarkably, we notice that the results of both DRAEM and NSA show significant performance improvements when integrating SSMCTB. Moreover, the performance gain brought by SSMCTB is always higher than the gain brought by SSPCAB. When taking advantage of the 3D nature of the MRI scans by employing the 3D SSMCTB, we attain even higher performance with DRAEM.

In Figure D.5, we present several examples of qualitative results from BRATS, given by DRAEM [56], before and after adding SSMCTB. In general, the localization

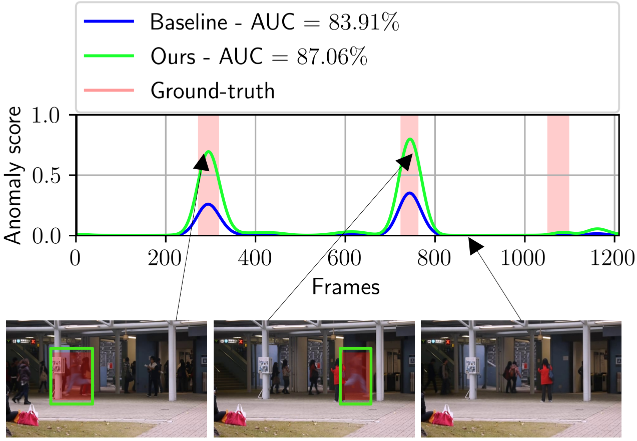


Fig. D.6: Frame-level anomaly scores of the method of Georgescu *et al.* [58], before (baseline) and after (ours) integrating SSMCTB, for test video 02 from the Avenue data set. Anomaly localization results correspond to the model based on SSMCTB. Best viewed in color.

results based on SSMCTB exhibit a higher overlap with the ground-truth regions, explaining why SSMCTB leads to superior performance levels.

4.6 Anomaly Detection in Videos

Baselines. We select five recent methods [42, 44, 55, 58, 59] yielding state-of-the-art performance on Avenue and ShanghaiTech. Liu *et al.* [42] proposed a GAN-based framework to detect anomalies based on the future frame prediction error. Park *et al.* [44] presented a memory-based auto-encoder classifying anomalies based on the reconstruction error. The model comprises a memory module that memorizes prototypes of normal samples. Liu *et al.* [59] employed a hybrid framework based on flow reconstruction and frame prediction, using the accumulated error to detect anomalies. Georgescu *et al.* [58] introduced a training scheme where the latent subspaces of appearance and motion auto-encoders are improved by performing gradient ascent on pseudo-anomalies during training. Bărbălău *et al.* [55] extended the previous work of Georgescu *et al.* [50] with two 3D transformer-based self-supervised multi-task architectures trained on new sets of proxy tasks. Among the two versions proposed in [55], we opt for SSMTL++v2. We included this 3D model [55] because it serves as a good baseline for applying our 3D SSMCTB.

We also experiment with the recently proposed masked auto-encoder framework [60], which is based on the ViT backbone [52]. We add this baseline model to demonstrate the applicability of SSMCTB to vision transformers.

Results on RGB videos. We present the results on Avenue and ShanghaiTech in Table D.5. As for the image anomaly detection experiments, we compare the results of the underlying models before and after adding SSPCAB [54] and SSMCTB, respectively. For the method of Liu *et al.* [42], both SSPCAB and SSMCTB lead to perfor-

Table D.5: Micro-averaged frame-level AUC, macro-averaged frame-level AUC, RBDC, and TBDC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech. Among the existing models, we select six models [42, 44, 55, 58–60] to show results before and after including SSPCAB and SSMCTB, respectively. The best result for each underlying model is highlighted in bold. The top score for each metric is shown in red.

Method	Avenue				ShanghaiTech			
	AUC		RBDC	TBDC	AUC		RBDC	TBDC
	Micro	Macro			Micro	Macro		
Liu *et al.* [13]	84.4	-	-	-	-	-	-	-
Sultani *et al.* [102]	-	-	-	-	-	76.5	-	-
Ionescu *et al.* [18]	88.9	-	-	-	-	-	-	-
Nguyen *et al.* [43]	86.9	-	-	-	-	-	-	-
Ionescu *et al.* [17]	87.4	90.4	15.77	27.01	78.7	84.9	20.65	44.54
Wu *et al.* [103]	86.6		-	-	-	-	-	-
Lee *et al.* [104]	90.0		-	-	-	-	-	-
Yu *et al.* [89]	89.6	-	-	-	74.8	-	-	-
Ramachandra *et al.* [19]	72.0		35.80	80.90	-	-	-	-
Ramachandra *et al.* [20]	87.2		41.20	78.60	-	-	-	-
Tang *et al.* [47]	85.1		-	-	73.0		-	-
Dong *et al.* [105]	84.9		-	-	73.7		-	-
Doshi *et al.* [106]	86.4		-	-	71.6		-	-
Sun *et al.* [107]	89.6		-	-	74.7		-	-
Wang *et al.* [108]	87.0		-	-	79.3		-	-
Astrid *et al.* [84]	84.7	-	-	-	73.7	-	-	-
Astrid *et al.* [109]	87.1	-	-	-	75.9	-	-	-
Georgescu *et al.* [50]	91.5	92.8	57.00	58.30	82.4	90.2	42.80	83.90
Liu *et al.* [42]	85.1	81.7	19.59	56.01	72.8	80.6	17.03	54.23
Liu *et al.* [42] + SSPCAB [54]	87.3	84.5	20.13	62.30	74.5	82.9	18.51	60.22
Liu *et al.* [42] + SSMCTB (Ours)	**89.5**	**84.6**	**23.79**	**66.03**	**74.6**	**83.9**	**19.13**	**61.65**
He *et al.* [60]	84.0	85.6	-	-	74.3	81.1	-	-
He *et al.* [60] + SSPCAB [54]	85.1	85.8	-	-	74.5	**81.9**	-	-
He *et al.* [60] + SSMCTB (Ours)	**86.4**	**86.5**	-	-	**76.1**	81.6	-	-
Park *et al.* [44]	82.8	86.8	-	-	68.3	79.7	-	-
Park *et al.* [44] + SSPCAB [54]	84.8	**88.6**	-	-	69.8	80.2	-	-
Park *et al.* [44] + SSMCTB (Ours)	**87.0**	87.7	-	-	**70.6**	**80.3**	-	-
Liu *et al.* [59]	89.9	93.5	41.05	86.18	74.2	83.2	44.41	83.86
Liu *et al.* [59] + SSPCAB [54]	**90.9**	92.2	**62.27**	**89.28**	**75.5**	83.7	45.45	84.50
Liu *et al.* [59] + SSMCTB (Ours)	89.6	**93.9**	46.49	86.43	75.2	**83.8**	**45.86**	**84.69**
Georgescu *et al.* [58]	92.3	90.4	65.05	**66.85**	82.7	89.3	**41.34**	78.79
Georgescu *et al.* [58] + SSPCAB [54]	92.9	**91.9**	65.99	64.91	**83.6**	**89.5**	40.55	**83.46**
Georgescu *et al.* [58] + SSMCTB (Ours)	**93.2**	91.8	**66.04**	65.12	83.3	**89.5**	40.52	81.93
Bărbălău *et al.* [55]	91.6	**92.5**	47.83	85.26	**83.8**	90.5	47.14	85.61
Bărbălău *et al.* [55] + 3D SSMCTB (Ours)	91.6	92.4	**49.01**	**85.94**	83.7	**90.6**	**47.73**	**85.68**

Table D.6: Micro and macro AUC scores (in %) on Thermal Rare Event, obtained while alternatively including SSPCAB [54] and SSMCTB into the method of Park *et al.* [44].

Method	AUC	
	Micro	Macro
Park *et al.* [44]	53.2	66.5
Park *et al.* [44] + SSPCAB	53.6	**66.6**
Park *et al.* [44] + SSMCTB (Ours)	**58.9**	**66.6**

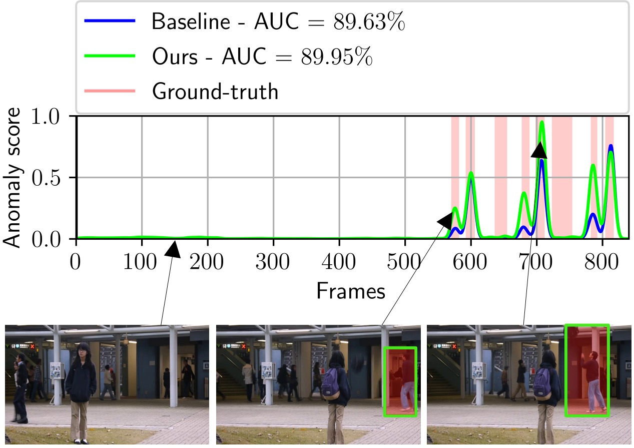


Fig. D.7: Frame-level anomaly scores of the method of Georgescu *et al.* [58], before (baseline) and after (ours) integrating SSMCTB, for test video 10 from the Avenue data set. Anomaly localization results correspond to the model based on SSMCTB. Best viewed in color.

mance improvements, but the gains brought by SSMCTB are always higher than those brought by SSPCAB. Since the methods of He *et al.* [60] and Park *et al.* [44] are only capable of detecting anomalies at the frame level, we only report their frame-level micro and macro AUC scores. The vanilla masked auto-encoder obtains competitive results on both Avenue and ShanghaiTech. On Avenue, SSMCTB brings higher gains to the masked auto-encoder than SSPCAB. On ShanghaiTech, SSMCTB is better than SSPCAB in terms of the micro AUC, but SSPCAB exhibits higher macro AUC gains. In summary, both SSMCTB and SSPCAB improve the masked auto-encoder, with SSMCTB having the upper hand. Considering the results of Park *et al.* [44] on Avenue, SSMCTB leads to higher gains in terms of the micro AUC (from 82.8% to 87.0%), while SSPCAB leads to a higher macro AUC (from 86.8% to 88.6%). On ShanghaiTech, we observe higher gains after adding SSMCTB rather than SSPCAB. Moving on to the object-centric models of Liu *et al.* [59] and Georgescu *et al.* [58], we observe that the top gains are mainly shared between SSPCAB and SSMCTB. When integrating our 3D SSMCTB into the 3D architecture presented in [55], we observe performance improvements according to most metrics. Overall, SSMCTB leads to the highest performance levels on Avenue for three metrics, namely the micro AUC (93.2%), the macro AUC (93.9) and the RBDC (66.04%). At the same time,

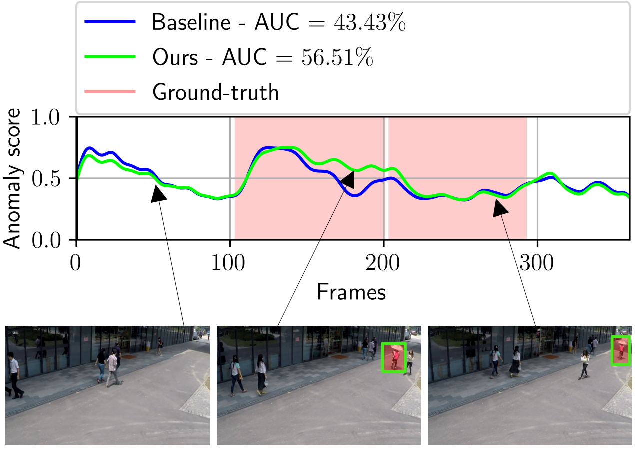


Fig. D.8: Frame-level anomaly scores of the method of Liu *et al.* [59], before (baseline) and after (ours) integrating SSMCTB, for test video 02_0164 from the ShanghaiTech data set. Anomaly localization results correspond to the model based on SSMCTB. Best viewed in color.

SSPCAB attains the highest TBDC score (89.28%) on Avenue. On ShanghaiTech, it appears that the best scores are obtained by adding the 3D SSMCTB into the underlying model of Bărbălău *et al.* [55], since our 3D SSMCTB brings performance gains for three metrics.

In Figure D.6 and D.7, we illustrate the anomaly detection performance on two test videos from Avenue, before and after integrating SSMCTB into the model of Georgescu *et al.* [58]. Our approach produces superior frame-level anomaly scores, being able to detect the person running in the first video (Figure D.6) and the person throwing an object in the second one (Figure D.7). Moreover, in the second video, we also notice that SSMCTB increases the anomaly score for the penultimate abnormal event, resolving the false negative detection of the baseline. Similarly, in Figure D.8, we show the effect of adding SSMCTB into the architecture of Liu *et al.* [59] applied on a test video from ShanghaiTech. Once again, SSMCTB improves the frame-level detection performance, being able to detect the person riding a bike in a pedestrian area, which is forbidden. SSMCTB correctly raises the anomaly scores for about 50 video frames, starting at around frame index 150, thus reducing the false negative rate.

Results on thermal videos. Since texture is not present in the thermal domain, there is no need to apply very deep architectures, as noticed by Nikolov *et al.* [62]. Moreover, object detectors pre-trained on natural images do not work equally well in the thermal domain due to the distribution shift. To this end, the object-centric [55, 58, 59] and very deep [42] baselines attain very poor results (micro AUC values under 50%). Hence, we resort to employing the architecture of Park *et al.* [44] as underlying model for SSPCAB and SSMCTB. As shown in Table D.6, the chosen baseline attains a micro AUC of 53.2% and a macro AUC of 66.5%. Both SSPCAB and SSMCTB seem to have a positive influence on the micro AUC score, but the gains of the latter block

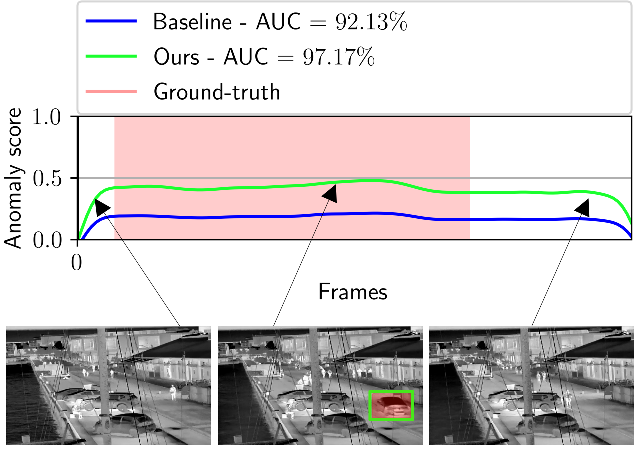


Fig. D.9: Frame-level anomaly scores of the method of Park *et al.* [44], before (baseline) and after (ours) integrating SSMCTB, for test video 39 from the Thermal Rare Event data set. Anomaly localization results correspond to the model based on SSMCTB. Best viewed in color.

Table D.7: Inference time (in milliseconds) per example for three frameworks [42, 56, 58], before and after integrating SSPCAB and SSMCTB, respectively. The running times are measured on an Nvidia GeForce GTX 3090 GPU with 24 GB of VRAM.

Method	Time (ms)			
	Baseline	+SSPCAB	+SSMCTB	+3D SSMCTB
Liu *et al.* [42]	2.1	2.4	2.5	-
Georgescu *et al.* [58]	1.5	1.7	1.8	-
Zavrtanik *et al.* [56]	26.4	-	-	26.6

are significantly higher (above 5%). In summary, the results reported on Thermal Rare Event demonstrate the utility of SSMCTB, further confirming the gains observed on RGB video data sets.

In Figure D.9, we show the anomaly detection performance on a test video from Thermal Rare Event, before and after integrating SSMCTB into the model of Park *et al.* [44]. SSMCTB leads to important gains in terms of the frame-level scores, being able to detect the vehicle moving backwards.

Inference time. Regardless of the underlying framework [42, 44, 55–60], similar to Ristea *et al.* [54], we add only one instance of SSMCTB, usually replacing the penultimate convolutional layer. Considering that the channel attention from SSPCAB is replaced with a channel-wise transformer block in SSMCTB, we might expect a slightly higher processing time. To assess the amount of extra time added by SSMCTB, we

Table D.8: Micro AUC (in %) on Avenue by incorporating SSMCTB into different conv blocks of the decoder proposed by Park *et al.* [44]. Along with the block placement, we also vary the dilation rate *d*.

Method	Decoder Conv Block	*d*	Micro AUC
Park *et al.* [44]	-	-	82.8
+SSMCTB	early	0	83.6
	early	1	83.2
	early	2	84.6
	early	3	84.7
	early	4	84.9
	middle	0	84.3
	middle	1	83.2
	middle	2	84.5
	middle	3	85.9
	middle	4	85.7
	late	0	86.2
	late	1	**87.0**
	late	2	85.9
	late	3	84.6
	late	4	85.8
	all	4,3,1	85.1

present the running times before and after integrating SSPCAB and SSMCTB into two state-of-the-art frameworks [42, 58] in Table D.7. For both baseline models, the time added by SSMCTB is at most 0.1 ms higher than the time taken by SSPCAB. Moreover, the computational time of SSMCTB does not exceed a difference of 0.4 ms with respect to the original baselines. Another important question is how does the 3D version of SSMCTB impact the running time. To answer this question, we take the DRAEM model [56] and measure the running time before and after adding the 3D SSMCTB. The reported time measurements show that the running time increase due to the 3D SSMCTB is still marginal, being around 0.2 ms. Hence, the processing delays caused by the introduction of the 2D or 3D SSMCTB versions are within the same range. In summary, we consider that the accuracy gains brought by SSMCTB outweigh the marginal running time expansions reported in Table D.7.

4.7 Ablation Study

Block placement. Across all the experiments presented so far, recall that we introduce a single SSMCTB, which is usually placed near the end of the architecture (penultimate convolutional layer), as mentioned in Section 4.3. The number of blocks as well as their placement should be tuned on some validation set, which could lead to higher performance gains. However, anomaly detection data sets do not commonly contain a validation set and there is no way to keep a number of training samples for validation, as the training set comprises only normal examples. To this end, we employed a single configuration (one block, closer to the output) to fairly demonstrate the universality of SSMCTB. Certainly, this choice might not always be optimal. Hence, we perform ablation experiments by incorporating SSMCTB at different decoder levels of the network proposed by Park *et al.* [44], considering different dilation rates (d). We vary the dilation rate along with the block placement, because Duţă *et al.* [110] observed that higher dilation rates are suitable for earlier dilated convolutional layers, and lower dilation rates are suitable for dilated convolutional layers closer to the output.

In Table D.8, we show the corresponding results on the Avenue data set. We start by adding SSMCTB into the earliest stage of the decoder (first conv block), progressively moving the block to the layers closer to the output of the decoder, until we reach the very last one. For each decoder level (early, middle, late), we vary the dilation rate to find a suitable value. We attain the best micro AUC (87.0%) when integrating SSMCTB into the last conv block of the decoder, while using a dilation rate of $d = 1$. A dilation rate of $d = 4$ seems suitable when placing SSMCTB at an earlier stage, while, for the middle stage placement, the optimal dilation rate appears to be $d = 3$. Interestingly, these results are consistent with the observation made by Duţă *et al.* [110], although their observation applies to dilated convolutions, while ours applies to masked convolutions. Nevertheless, all the results are consistently better than the baseline (82.8%), regardless of the block placement or the dilation rate. We do not observe major improvements when integrating multiple blocks, concluding that integrating a single SSMCTB is sufficient.

Table D.9: Micro AUC (in %) on Avenue by incorporating SSMCTB into the model of Park *et al.* [44], while varying the size of the masked region $\boldsymbol{M}$.

Method	Size of $\boldsymbol{M}$	Micro AUC
Park *et al.* [44]	-	82.8
+SSMCTB	1×1	87.0
	2×2	85.6
	3×3	84.9

Size of masked region. Increasing the size of the masked region $\boldsymbol{M}$ can lead to a harder reconstruction task, at each location where our masked convolution is applied. However, it is unclear if making the task harder leads to better results. To this end, we vary the spatial size of $\boldsymbol{M}$, considering three options: 1×1, 2×2 and 3×3. We present the corresponding results in Table D.9. The empirical results indicate that increasing the size of $\boldsymbol{M}$ leads to lower anomaly detection scores. Hence, we conclude that a size of 1×1 for the masked region $\boldsymbol{M}$ is optimal.

Transformer architecture. In Table D.10, we present further ablation experiments for the channel-wise transformer module. We keep the underlying model of Park *et al.* [44] and report the results on the Avenue data set. As variations for the transformer module, we consider the following hyperparameters: the activation map size ($h' \times w'$) after the average pooling layer, the token size (d_t) after the projection layer, the number of heads (H), as well as the number of successive transformer blocks (L).

First, we analyze how activation maps of different dimensions, given as output by the average pooling layer placed right before the transformer, influence the results.

Table D.10: Micro AUC (in %) on Avenue by incorporating SSMCTB into the model of Park *et al.* [44], while varying the hyperparameters of the channel-wise transformer, namely the activation map size ($h' \times w'$) after the average pooling layer, the token size (d_t) after the projection layer, the number of heads (H), as well as the number of successive transformer blocks (L).

Method	$h' \times w'$	d_t	H	L	Micro AUC
Park *et al.* [44]	-	-	-	-	82.8
+SSMCTB	1×1	64	4	2	87.0
	2×2	64	4	2	85.3
	3×3	64	4	2	85.2
	4×4	64	4	2	85.6
	1×1	16	4	2	84.6
	1×1	32	4	2	85.6
	1×1	64	4	2	87.0
	1×1	128	4	2	85.1
	1×1	64	3	2	85.6
	1×1	64	4	2	87.0
	1×1	64	5	2	87.0
	1×1	64	6	2	84.8
	1×1	64	4	1	85.1
	1×1	64	4	2	87.0
	1×1	64	4	3	84.0

We observe that shrinking the maps to 1×1 gives the best micro AUC (87.0%). The optimal configuration of the average pooling layer (producing activation maps of 1×1) is equivalent to global average pooling. For the projection layer, we consider output dimensions in the set $d_t \in \{16, 32, 64, 128\}$. The optimal size for the projection layer is $d_t = 64$. We consider transformer modules having 3 to 6 heads. The empirical evidence indicates that using $H = 4$ or $H = 5$ heads leads to equally good results. Finally, we experiment with transformer modules having 1 to 3 blocks. The best performance is achieved with $L = 2$ successive transformer blocks. We underline that all transformer configurations surpass the baseline model [44].

Huber loss hyperparameter. Huber loss is the combination of the L_1 (MAE) and L_2 (MSE) losses (see Eq. (D.7)), where δ is a hyperparameter representing the threshold that switches between the two loss functions. To study the effect of δ, we consider different values for the hyperparameter $\delta \in \{0.5, 1, 2\}$, reporting the results in Table D.11. We find that the maximum improvement corresponds to $\delta = 1$, but the other values of δ also lead to superior results compared to the baseline.

Comparison with dilated convolution. In Table D.12, we compare the dilated convolution against the proposed masked convolution, alternating between the two operations inside SSMCTB. We denote the block based on dilated convolution through the acronym SSDCTB. When comparing the two convolutional operations, we consider multiple dilation rates between 1 and 3. The experiments show that the proposed masked convolution outperforms the dilated convolution, regardless of the dilation rate. This confirms that the two operations are not equivalent, essentially revealing the

Table D.11: Micro AUC (in %) on Avenue by incorporating SSMCTB into the model of Park *et al.* [44], while varying the hyperparameter δ of the Huber loss.

Method	δ	Micro AUC
Park *et al.* [44]	-	82.8
+SSMCTB	0.5	84.1
	1	87.0
	2	85.8

Table D.12: Micro AUC (in %) on Avenue by incorporating SSMCTB into the model of Park *et al.* [44], while switching between dilated and masked convolution. Different values for the dilation rate *d* are tested for the two operations.

Method	*d*	Micro AUC
Park *et al.* [44]	-	82.8
+SSDCTB (dilated conv)	1	85.1
	2	83.3
	3	85.0
+SSMCTB (masked conv)	1	87.0
	2	85.5
	3	85.9

importance of the self-supervised task based on reconstructing the masked region $\boldsymbol{M}$ situated in the center of the receptive field.

5 Conclusion

In this paper, we extended our previous work [54] by introducing SSMCTB, a novel neural block composed of a masked convolutional layer and a channel-wise transformer module, which predicts a masked region in the center of the convolutional receptive field. Our neural block is trained in a self-supervised manner, via a reconstruction loss of its own. To show the benefits of using SSMCTB in anomaly detection, we integrated our block into a series of image and video anomaly detection methods [42, 44, 55–60]. In addition, we included two new benchmarks from domains that were not previously considered by Ristea *et al.* [54], namely medical images and thermal videos. Moreover, we extended the 2D masked convolution to a 3D masked convolution, broadening the applicability of the self-supervised block to 3D neural architectures. To showcase the utility of the new 3D SSMCTB, we integrated our 3D block into two 3D networks (3D DRAEM and SSMTL++v2) for anomaly detection in image and video, respectively. Our empirical results across multiple benchmarks and underlying models indicate that SSMCTB brings performance improvements in a vast majority of cases. Furthermore, with the help of SSMCTB, we are able to obtain new state-of-the-art levels on the widely-used Avenue and ShanghaiTech data sets. We consider this as a major achievement, which would not have been possible without

SSMCTB.

In future work, we aim to apply our novel self-supervised block on other tasks, aside from anomaly detection. For example, due to the self-supervised loss computed with respect to the masked region, our block could be integrated into various neural architectures to perform self-supervised pre-training, before applying the respective models to downstream tasks. Interestingly, the pre-training could be performed at multiple architectural levels, *i.e.* wherever the block is added into the model.

Acknowledgment

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS - UEFISCDI, project no. PN-III-P2-2.1-PED-2021-0195, contract no. 690/2022, within PNCDI III. The research leading to these results has also received funding from the NO Grants 2014-2021, under project ELO-Hyp contract no. 24/2020. Additionally, this work has been funded by Milestone Systems through the Milestone Research Programme at AAU, and by SecurifAI.

References

[1] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, “MVTec AD – A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” in *Proceedings of CVPR*, 2019, pp. 9592–9600.

[2] S. Lee, S. Lee, and B. C. Song, “CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization,” *IEEE Access*, vol. 10, pp. 78 446–78 454, 2022.

[3] W. Luo, W. Liu, and S. Gao, “A Revisit of Sparse Coding Based Anomaly Detection in Stacked RNN Framework,” in *Proceedings of ICCV*, 2017, pp. 341–349.

[4] N. Shvetsova, B. Bakker, I. Fedulova, H. Schulz, and D. V. Dylov, “Anomaly Detection in Medical Imaging With Deep Perceptual Autoencoders,” *IEEE Access*, vol. 9, pp. 118 571–118 583, 2021.

[5] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone, “Defect Detection in SEM Images of Nanofibrous Materials,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 551–561, 2017.

[6] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, “Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression,” in *Proceedings of CVPR*, 2015, pp. 2909–2917.

[7] Y. Cong, J. Yuan, and J. Liu, “Sparse reconstruction cost for abnormal event detection,” in *Proceedings of CVPR*, 2011, pp. 3449–3456.

[8] J. K. Dutta and B. Banerjee, “Online Detection of Abnormal Events Using Incremental Coding Length,” in *Proceedings of AAAI*, 2015, pp. 3755–3761.

[9] C. Lu, J. Shi, and J. Jia, “Abnormal Event Detection at 150 FPS in MATLAB,” in *Proceedings of ICCV*, 2013, pp. 2720–2727.

[10] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund, “Unsupervised Behavior-Specific Dictionary Learning for Abnormal Event Detection,” in *Proceedings of BMVC*, 2015, pp. 28.1–28.13.

[11] A. Del Giorno, J. Bagnell, and M. Hebert, “A Discriminative Framework for Anomaly Detection in Large Videos,” in *Proceedings of ECCV*, 2016, pp. 334–349.

[12] R. T. Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, “Unmasking the abnormal events in video,” in *Proceedings of ICCV*, 2017, pp. 2895–2903.

[13] Y. Liu, C.-L. Li, and B. Póczos, “Classifier Two-Sample Test for Video Anomaly Detections,” in *Proceedings of BMVC*, 2018.

[14] G. Pang, C. Yan, C. Shen, A. v. d. Hengel, and X. Bai, "Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection," in *Proceedings of CVPR*, 2020, pp. 12 173–12 182.

[15] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Uninformed Students: Student-Teacher Anomaly Detection With Discriminative Latent Embeddings," in *Proceedings of CVPR*, 2020, pp. 4183–4192.

[16] T. Defard, A. Setkov, A. Loesch, and R. Audigier, "PaDiM: A patch distribution modeling framework for anomaly detection and localization," in *Proceedings of ICPR*, 2021, pp. 475–489.

[17] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, "Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video," in *Proceedings of CVPR*, 2019, pp. 7842–7851.

[18] R. T. Ionescu, S. Smeureanu, M. Popescu, and B. Alexe, "Detecting abnormal events in video using Narrowed Normality Clusters," in *Proceedings of WACV*, 2019, pp. 1951–1960.

[19] B. Ramachandra and M. Jones, "Street Scene: A new dataset and evaluation protocol for video anomaly detection," in *Proceedings of WACV*, 2020, pp. 2569–2578.

[20] B. Ramachandra, M. Jones, and R. Vatsavai, "Learning a distance function with a Siamese network to localize anomalies in videos," in *Proceedings of WACV*, 2020, pp. 2598–2607.

[21] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-Play CNN for Crowd Motion Analysis: An Application in Abnormal Event Detection," in *Proceedings of WACV*, 2018, pp. 1689–1698.

[22] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-Cascade: Cascading 3D Deep Neural Networks for Fast Anomaly Detection and Localization in Crowded Scenes," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1992–2004, 2017.

[23] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[24] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proceedings of CVPR*, 2012, pp. 2112–2119.

[25] S. Smeureanu, R. T. Ionescu, M. Popescu, and B. Alexe, "Deep Appearance Features for Abnormal Behavior Detection in Video," in *Proceedings of ICIAP*, vol. 10485, 2017, pp. 779–789.

[26] Q. Sun, H. Liu, and T. Harada, “Online growing neural gas for anomaly detection in changing surveillance scenes,” *Pattern Recognition*, vol. 64, no. C, pp. 187–201, Apr. 2017.

[27] H. T. Tran and D. Hogg, “Anomaly Detection using a Convolutional Winner-Take-All Autoencoder,” in *Proceedings of BMVC*, 2017.

[28] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, “Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[29] B. Antic and B. Ommer, “Video parsing for abnormality detection,” in *Proceedings of ICCV*, 2011, pp. 2415–2422.

[30] Y. Feng, Y. Yuan, and X. Lu, “Learning deep event models for crowd anomaly detection,” *Neurocomputing*, vol. 219, pp. 548–556, 2017.

[31] R. Hinami, T. Mei, and S. Satoh, “Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge,” in *Proceedings of ICCV*, 2017, pp. 3639–3647.

[32] J. Kim and K. Grauman, “Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates,” in *Proceedings of CVPR*, 2009, pp. 2921–2928.

[33] V. Mahadevan, W.-X. LI, V. Bhalodia, and N. Vasconcelos, “Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 1975–1981.

[34] R. Mehran, A. Oyama, and M. Shah, “Abnormal crowd behavior detection using social force model,” in *Proceedings of CVPR*, 2009, pp. 935–942.

[35] M. Rudolph, B. Wandt, and B. Rosenhahn, “Same Same But DifferNet: Semi-Supervised Defect Detection with Normalizing Flows,” in *Proceedings of WACV*, 2021, pp. 1907–1916.

[36] B. Saleh, A. Farhadi, and A. Elgammal, “Object-Centric Anomaly Detection by Attribute-Based Reasoning,” in *Proceedings of CVPR*, 2013, pp. 787–794.

[37] S. Wu, B. E. Moore, and M. Shah, “Chaotic Invariants of Lagrangian Particle Trajectories for Anomaly Detection in Crowded Scenes,” in *Proceedings of CVPR*, 2010, pp. 2054–2060.

[38] Y. Fei, C. Huang, C. Jinkun, M. Li, Y. Zhang, and C. Lu, “Attribute Restoration Framework for Anomaly Detection,” *IEEE Transactions on Multimedia*, vol. 24, pp. 116–127, 2022.

[39] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, “Memorizing Normality to Detect Anomaly: Memory-Augmented Deep Autoencoder for Unsupervised Anomaly Detection,” in *Proceedings of ICCV*, 2019, pp. 1705–1714.

[40] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in *Proceedings of CVPR*, 2016, pp. 733–742.

[41] Z. Li, N. Li, K. Jiang, Z. Ma, X. Wei, X. Hong, and Y. Gong, “Superpixel Masking and Inpainting for Self-Supervised Anomaly Detection,” in *Proceedings of BMVC*, 2020.

[42] W. Liu, W. Luo, D. Lian, and S. Gao, “Future Frame Prediction for Anomaly Detection – A New Baseline,” in *Proceedings of CVPR*, 2018, pp. 6536–6545.

[43] T.-N. Nguyen and J. Meunier, “Anomaly Detection in Video Sequence With Appearance-Motion Correspondence,” in *Proceedings of ICCV*, 2019, pp. 1273–1283.

[44] H. Park, J. Noh, and B. Ham, “Learning Memory-guided Normality for Anomaly Detection,” in *Proceedings of CVPR*, 2020, pp. 14 372–14 381.

[45] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, “Abnormal Event Detection in Videos using Generative Adversarial Nets,” in *Proceedings of ICIP*, 2017, pp. 1577–1581.

[46] M. Salehi, N. Sadjadi, S. Baselizadeh, M. H. Rohban, and H. R. Rabiee, “Multiresolution Knowledge Distillation for Anomaly Detection,” in *Proceedings of CVPR*, 2021, pp. 14 902–14 912.

[47] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, “Integrating prediction and reconstruction for anomaly detection,” *Pattern Recognition Letters*, vol. 129, pp. 123–130, 2020.

[48] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, “Attention guided anomaly localization in images,” in *Proceedings of ECCV*, 2020, pp. 485–503.

[49] P. J. Huber, “Robust Estimation of a Location Parameter,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[50] M.-I. Georgescu, A. Bărbălău, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “Anomaly Detection in Video via Self-Supervised and Multi-Task Learning,” in *Proceedings of CVPR*, 2021, pp. 12 742–12 752.

[51] A. Acsintoae, A. Florescu, M. Georgescu, T. Mare, P. Sumedrea, R. T. Ionescu, F. S. Khan, and M. Shah, "Ubnormal: New benchmark for supervised open-set video anomaly detection," in *Proceedings of CVPR*, 2022, pp. 20 143–20 153.

[52] 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," in *Proceedings of ICLR*, 2021.

[53] 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 *Proceedings of NIPS*, vol. 30, 2017.

[54] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection," in *Proceedings of CVPR*, 2022, pp. 13 576–13 586.

[55] A. Bărbălău, R. T. Ionescu, M.-I. Georgescu, J. Dueholm, B. Ramachandra, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "SSMTL++: Revisiting Self-Supervised Multi-Task Learning for Video Anomaly Detection," *Computer Vision and Image Understanding*, vol. 229, p. 103656, 2023.

[56] V. Zavrtanik, M. Kristan, and D. Skocaj, "DRAEM – A Discriminatively Trained Reconstruction Embedding for Surface Anomaly Detection," in *Proceedings of ICCV*, 2021, pp. 8330–8339.

[57] H. M. Schlüter, J. Tan, B. Hou, and B. Kainz, "Natural synthetic anomalies for self-supervised anomaly detection and localization," in *Proceedings of ECCV*, 2022.

[58] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 9, pp. 4505–4523, 2022.

[59] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A Hybrid Video Anomaly Detection Framework via Memory-Augmented Flow Reconstruction and Flow-Guided Frame Prediction," in *Proceedings of ICCV*, 2021, pp. 13 588–13 597.

[60] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," in *Proceedings of CVPR*, 2022, pp. 16 000–16 009.

[61] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M.-A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, C. Demiralp, C. R. Durst,

M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. S. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H.-C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput, “The Multimodal Brain Tumor Image Segmentation Benchmark (BraTS),” *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, 2015.

[62] 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 *Proceedings of NeurIPS*, 2021.

[63] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *Proceedings of CVPR*, 2018, pp. 7132–7141.

[64] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proceedings of ECCV*, 2020, pp. 213–229.

[65] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, “TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation,” *arXiv preprint arXiv:2102.04306*, 2021.

[66] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in Vision: A Survey,” *ACM Computing Surveys*, 2021.

[67] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, “Image transformer,” in *Proceedings of ICML*, 2018, pp. 4055–4064.

[68] N.-C. Ristea, A.-I. Miron, O. Savencu, M.-I. Georgescu, N. Verga, F. S. Khan, and R. T. Ionescu, “CyTran: Cycle-Consistent Transformers for Non-Contrast to Contrast CT Translation,” *arXiv preprint arXiv:2110.06400*, 2021.

[69] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *Proceedings of ICML*, 2021, pp. 10 347–10 357.

[70] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, “CvT: Introducing Convolutions to Vision Transformers,” in *Proceedings of ICCV*, 2021, pp. 22–31.

[71] X. Xu and N. Xu, “Hierarchical Image Generation via Transformer-Based Sequential Patch Selection,” in *Proceedings of AAAI*, 2022, pp. 2938–2945.

[72] B. Zhang, S. Gu, B. Zhang, J. Bao, D. Chen, F. Wen, Y. Wang, and B. Guo, "StyleSwin: Transformer-based GAN for High-resolution Image Generation," in *Proceedings of CVPR*, 2022, pp. 11 304–11 314.

[73] M. Zheng, P. Gao, X. Wang, H. Li, and H. Dong, "End-to-end object detection with adaptive clustering transformer," in *Proceedings of BMVC*, 2020.

[74] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Proceedings of ICLR*, 2020.

[75] J. Jiang, J. Zhu, M. Bilal, Y. Cui, N. Kumar, R. Dou, F. Su, and X. Xu, "Masked Swin Transformer Unet for Industrial Anomaly Detection," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 2200–2209, 2023.

[76] Y. Lee and P. Kang, "AnoViT: Unsupervised anomaly detection and localization with vision transformer-based encoder-decoder," *IEEE Access*, vol. 10, pp. 46 717–46 724, 2022.

[77] P. Mishra, R. Verk, D. Fornasier, C. Piciarelli, and G. L. Foresti, "VT-ADL: A vision transformer network for image anomaly detection and localization," in *Proceedings of ISIE*. IEEE, 2021, pp. 1–6.

[78] J. Pirnay and K. Chai, "Inpainting transformer for anomaly detection," in *Proceedings of ICIAP*, 2022, pp. 394–406.

[79] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proceedings of ICLR*, 2022.

[80] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context Encoders: Feature Learning by Inpainting," in *Proceedings of CVPR*, 2016, pp. 2536–2544.

[81] C. Wei, H. Fan, S. Xie, C.-Y. Wu, A. Yuille, and C. Feichtenhofer, "Masked Feature Prediction for Self-Supervised Visual Pre-Training," in *Proceedings of CVPR*, 2022, pp. 14 668–14 678.

[82] H. Chang, H. Zhang, L. Jiang, C. Liu, and W. T. Freeman, "MaskGIT: Masked Generative Image Transformer," in *Proceedings of CVPR*, 2022, pp. 11 315–11 325.

[83] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *Proceedings of CVPR*, 2022, pp. 19 313–19 322.

[84] M. Astrid, M. Z. Zaheer, and S.-I. Lee, “Synthetic Temporal Anomaly Guided End-to-End Video Anomaly Detection,” in *Proceedings of ICCVW*, 2021, pp. 207–214.

[85] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. V. Gehler, “Towards Total Recall in Industrial Anomaly Detection,” in *Proceedings of CVPR*, 2022, pp. 14 298–14 308.

[86] S. Yamada, S. Kamiya, and K. Hotta, “Reconstructed Student-Teacher and Discriminative Networks for Anomaly Detection,” in *Proceedings of IROS*, 2022, pp. 2725–2732.

[87] K. Doshi and Y. Yilmaz, “Rethinking video anomaly detection - a continual learning approach,” in *Proceedings of WACV*, 2022, pp. 3961–3970.

[88] M. Haselmann, D. P. Gruber, and P. Tabatabai, “Anomaly detection using deep learning based image completion,” *Proceedings of ICMLA*, pp. 1237–1242, 2018.

[89] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, “Cloze Test Helps: Effective Video Anomaly Detection via Learning to Complete Video Events,” in *Proceedings of ACMMM*, 2020, pp. 583–591.

[90] M. Sabokrou, M. PourReza, M. Fayyaz, R. Entezari, M. Fathy, J. Gall, and E. Adeli, “AVID: Adversarial Visual Irregularity Detection,” in *Proceedings of ACCV*, 2018, pp. 488–505.

[91] D. Luo, C. Liu, Y. Zhou, D. Yang, C. Ma, Q. Ye, and W. Wang, “Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning,” in *Proceedings of AAAI*, 2020, pp. 11 701–11 708.

[92] 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 ICCV*, 2021, pp. 10 012–10 022.

[93] X. Guo, Z. Jin, C. Chen, H. Nie, J. Huang, D. Cai, X. He, and X. Hua, “Discriminative-Generative Dual Memory Video Anomaly Detection,” *arXiv preprint arXiv:2104.14430*, 2021.

[94] C. Li, K. Sohn, J. Yoon, and T. Pfister, “CutPaste: Self-Supervised Learning for Anomaly Detection and Localization,” in *Proceedings of CVPR*, 2021, pp. 9664–9674.

[95] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Proceedings of NIPS*, 2012, pp. 1106–1114.

[97] M. D. Zeiler and R. Fergus, “Visualizing and Understanding Convolutional Networks,” in *Proceedings of ECCV*, 2014, pp. 818–833.

[98] S. Sabour, N. Frosst, and G. E. Hinton, “Dynamic Routing Between Capsules,” in *Proceedings of NIPS*, 2017, pp. 3859–3869.

[99] A. Şandru, M.-I. Georgescu, and R. T. Ionescu, “Feature-level augmentation to improve robustness of deep neural networks to affine transformations,” in *Proceedings of ECCVW*, 2022.

[100] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of ICML*, 2010, pp. 807–814.

[101] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, “A Survey of Single-Scene Video Anomaly Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[102] W. Sultani, C. Chen, and M. Shah, “Real-World Anomaly Detection in Surveillance Videos,” in *Proceedings of CVPR*, 2018, pp. 6479–6488.

[103] P. Wu, J. Liu, and F. Shen, “A Deep One-Class Neural Network for Anomalous Event Detection in Complex Scenes,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 7, pp. 2609–2622, 2019.

[104] S. Lee, H. G. Kim, and Y. M. Ro, “BMAN: Bidirectional Multi-Scale Aggregation Networks for Abnormal Event Detection,” *IEEE Transactions on Image Processing*, vol. 29, pp. 2395–2408, 2019.

[105] F. Dong, Y. Zhang, and X. Nie, “Dual Discriminator Generative Adversarial Network for Video Anomaly Detection,” *IEEE Access*, vol. 8, pp. 88 170–88 176, 2020.

[106] K. Doshi and Y. Yilmaz, “Any-Shot Sequential Anomaly Detection in Surveillance Videos,” in *Proceedings of CVPRW*, 2020, pp. 934–935.

[107] C. Sun, Y. Jia, Y. Hu, and Y. Wu, “Scene-Aware Context Reasoning for Unsupervised Abnormal Event Detection in Videos,” in *Proceedings of ACMMM*, 2020, pp. 184–192.

[108] Z. Wang, Y. Zou, and Z. Zhang, “Cluster Attention Contrast for Video Anomaly Detection,” in *Proceedings of ACMMM*, 2020, pp. 2463–2471.

[109] M. Astrid, M. Z. Zaheer, J.-Y. Lee, and S.-I. Lee, “Learning not to reconstruct anomalies,” in *Proceedings of BMVC*, 2021.

[110] I. C. Duţă, M. I. Georgescu, and R. T. Ionescu, “Contextual convolutional neural networks,” in *Proceedings of ICCVW*, 2021, pp. 403–412.

Paper E

Temporal Cues from Socially Unacceptable Trajectories for Anomaly Detection

Neelu Madan, Arya Farkhondeh, Kamal Nasrollahi, Sergio Escalera,
Thomas B. Moeslund

The paper has been published in the
International Conference on Computer Vision Workshops (ICCVW), pp.2150-2158,
2021.

Abstract

State-of-the-Art (SoTA) deep learning-based approaches to detect anomalies in surveillance videos utilize limited temporal information, including basic information from motion, e.g., optical flow computed between consecutive frames. In this paper, we compliment the SoTA methods by including long-range dependencies from trajectories for anomaly detection. To achieve that, we first created trajectories by running a tracker on two SoTA datasets, namely Avenue and Shanghai-Tech. We propose a prediction-based anomaly detection method using trajectories based on Social GANs, also called in this paper as temporal-based anomaly detection. Then, we hypothesize that late fusion of the result of this temporal-based anomaly detection system with spatial-based anomaly detection systems produces SoTA results. We verify this hypothesis on two spatial-based anomaly detection systems. We show that both cases produce results better than baseline spatial-based systems, indicating the usefulness of the temporal information coming from the trajectories for anomaly detection. We observe that the proposed approach depicts the maximum improvement in micro-level Area-Under-the-Curve (AUC) by 4.1% on CUHK Avenue and 3.4% on Shanghai-Tech over one of the baseline method. We also show a high performance on cross-data evaluation, where we learn the weights to combine spatial and temporal information on Shanghai-Tech and perform evaluation on CUHK Avenue and vice-versa.

1 Introduction

Video anomaly detection is a sub-domain of behavior understanding, where anomalies for applications such as theft detection, traffic light jumping, and fighting, etc. are getting increasingly relevant with the accessibility and proliferation of video surveillance. There are multiple challenges associated with anomaly detection including the vague definition of anomalous behavior, i.e., anomaly changes with the context. An example to illustrate the context can be that driving a vehicle on a pedestrian street is considered anomalous while it is normal in the context of a road. Additionally, by definition anomalies are rare to anticipate, which consequently leads to the failure of supervised learning methods due to imbalanced datasets.

Therefore, unsupervised and weakly supervised anomaly detection approaches have recently gained interest. Common examples are reconstruction [1] and prediction [2] based anomaly detection. Reconstruction-based anomaly detection systems reconstruct the current frame and prediction-based ones predict the future frame. If the reconstruction/prediction error is low, the current/future frame is normal, otherwise abnormal. State-of-the-Art deep learning approaches for anomaly detection are only trained for normal events, with the hypothesis that the reconstruction/prediction error for anomalous frames is high. However, neural networks sometimes learn to reconstruct/predict even anomalous frames with low errors. This reduces the discriminative power of the neural network to classify a frame as abnormal or normal. To overcome this drawback, memory-augmented auto-encoders [3, 4] are proposed. The memory-augmented auto-encoders [3, 4] contain an extra memory module along with a prediction/reconstruction-based network. The memory module learns to cluster the normal events in the training data and finally uses a one-class classification approach to identify the anomalies. It basically creates a prototype for each normal event in the training data and prevents the network from generalizing for abnormal events. Despite the great achievements of SoTA methods in anomaly detection, still, there is room for improvements. SoTA approaches are mostly using spatial information for anomaly detection and utilizing temporal information has been limited to gradient or optical flow computed between consecutive frames. Obtaining the optical flow for large datasets is a time-consuming and computationally expensive process. This is the reason that most anomaly detection systems utilizing optical flow extract this from only two frames [5]. The object's trajectories, which implicitly include the history of motion [6] are better choices and are also computationally efficient. However, contextual anomalies such as walking in restricted zones and behavioral anomalies such as dancing or jumping are not captured by using only trajectories. Therefore, we need an appropriate balance of spatial and temporal information for robust anomaly detection.

In this paper, we hypothesize that fusing temporal anomaly detection scores (based on trajectories) with spatial anomaly detection scores (based on SoTA methods) increases the accuracy of these systems, regardless of the network architecture used for spatial anomaly detection. To encode the long-range dependencies for the video anomaly detection, we use these trajectories to detect the anomalies using our proposed temporal network based on Social Generative Adversarial Networks (Social GANs) [7]. We implicitly consider social interaction among different objects in the scene during anomaly detection using trajectories because of the presence of social pooling layer in Social GANs [7]. We verify our hypothesis by using different baselines, i.e., prediction-based system of Liu *et al.* [2] and memory-based system of Park *et al.* [3] for our spatial network. The prediction-based system of Liu *et al.* [2] predicts a future frame from the past four frames by minimizing intensity, gradient, and flow loss. However, the memory-based system of Park *et al.* [3] incorporates additional memory modules for both prediction-based and reconstruction-based anomaly detection. For the inclusion of temporal information from trajectories, we learn a score level fusion of anomaly detection scores obtained from the temporal and spatial networks.

We verify that there is improvement in frame-level AUC (a commonly used metric for video anomaly detection) for each baseline by using the complementary information from trajectories. There is an improvement of 1.7% on CUHK Avenue [8] and 1.8% on Shanghai-Tech [2] for Liu *et al.* [2]. The inclusion of trajectories in Park *et al.* [3] shows an improvement of 4.1% on CUHK Avenue [8] and 3.3% on Shanghai-Tech [2] for reconstruction-based and an improvement of 0.1% on CUHK Avenue [8] and 3.3% on Shanghai-Tech [2] for prediction-based approaches. We also perform some additional experiments on cross-database generalization, where we learn parameters on Shanghai-Tech [2] and use them to evaluate CUHK Avenue [8] or vice-versa. We observe an overall increase in performance even in-case of cross-databases experiments, i.e., from CUHK Avenue [8] to Shanghai-Tech [2] have an improvement of 0.7% and from Shanghai-Tech [2] to CUHK Avenue [8] have an improvement of 1.8% in the AUC over the baseline by Li *et al.* [2]. The late fusion of spatial and temporal information makes our approach applicable to any SoTA anomaly detection method.

2 Related Work

Systems to deal with the task of video anomaly detection are getting complex with the evolution of complex anomalies and new datasets. The methods use for video anomaly detections are broadly classified into two categories namely spatial-based and temporal-based anomaly detection.

2.1 Anomaly Detection Using Spatial Cues

Anomaly detection systems utilizing spatial information can be further classified into four sub-categories: Reconstruction, Prediction, Hybrid and Object-centric approaches. Reconstruction-based approaches seek to learn normalcy, where the expectation is that anomalous activity will have a large reconstruction error, comparing the input with its reconstruction. This approach has shown promise due to the era of deep learning and specifically the convolutional autoencoder (CAE) and the generative adversarial network (GAN) [9]. The work of Hasan *et al.* [1] is the first example of applying CAE and comparing it to hand-crafted features like Histograms of Oriented Gradients (HOG) and Histograms of Optical Flows (HOF), showing the potential of learned representations. Similar approach is seen using GANs [10, 11]. Prediction-based approaches argue that anomalous actions are naturally harder to predict. This approach is pioneered by Liu *et al.* [2], using a sliding time window to predict the future frame. The future prediction is then compared to the actual input. This is further expanded by Rodrigues *et al.* [12] using multiple timescales. Hybrid approaches [13] [14] [15] are combining both the reconstruction and prediction aspects. To avail the success of deep learning-based object-detection, few anomaly detection approaches such as [5, 16, 17] incorporates anomaly score based on object detection rather than on frame-level.

Training unsupervised methods for a complex task such as anomaly detection is challenging due to limited guidance during learning, compared to supervised learning. There are some methods that are adding some prior information to the above approaches for improving accuracy. A common approach to aid in the learning is to use pre-trained systems to impose what is already known and learned, either in the form of optical flow [5], object detectors [10, 16, 18], skeletons [14], or memory augmentation [3, 4]. The downside of many of these methods is the limited use of contextual information. In recent years, memory-augmentation networks that are using external memory to extend the capabilities of the neural network are used, e.g., Gong. *et al.* [4] proposed a memory-augmented deep autoencoder, where rather than reconstructing the frame directly, the representation obtained from the encoder part is used for querying the most relevant information out of the memory for reconstruction. These types of networks mitigated the issue that abnormal frames can also be reconstructed with a small error.

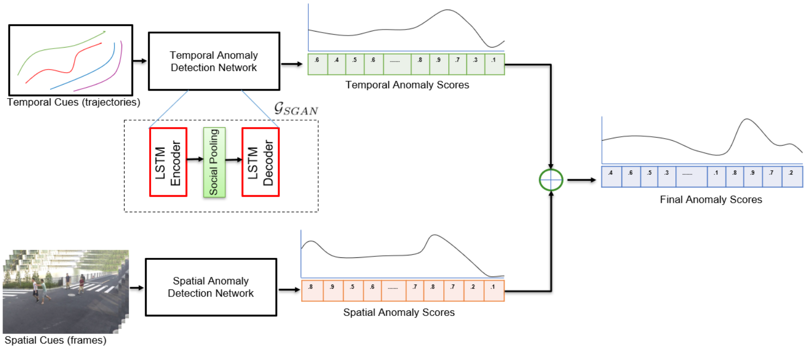


Fig. E.1: Proposed system for anomaly detection. It contains spatial and temporal branch, respectively. Weighted combination of spatial and temporal anomaly detection is used to generate the final anomaly score.

2.2 Anomaly Detection Using Temporal Cues

SoTA anomaly detection approaches are mostly using spatial cues, while taking only limited temporal information into consideration. For example, Liu *et al.* [2] use optical flow between consecutive frames, Ionescu *et al.* [16] use backward gradient between the previous and current frame and forward gradient between current and next frame. Later, Georgescu *et al.* [5] verified that optical flow is better to capture motion in the context of anomaly detection, so they replaced forward and backward gradient in Ionescu *et al.* [16] by forward and backward optical flow.

There are limited approaches such as Morias *et al.* [14] and Rodrigue *et al.* [12] including trajectory for anomaly detection. Morias *et al.* [14] uses a skeleton-based representation of trajectories, which needs additional annotations for gaits in human body. To further expand this work, Rodrigue *et al.* [12] also uses pose-based trajectories but extracted features at multiple scales. The limitation of posed-based trajectories is that they are only applicable for human anomalies, and non-human anomalies such as vehicles on the pedestrian street or unattended luggage cannot be detected.

Some examples of anomaly detection using trajectories on traffic and old datasets include [19, 20], which are based on the clustering of trajectories using hand-crafted features and distance measures between the trajectories. In this case, the clusters with small support are anomalous. Some other statistical approaches use for anomaly detection include probabilistic modeling and learning of normal trajectories, e.g., [21] applied Hidden Markov Model followed by K-Mean clustering. A rule-based classifier implemented by [22] applies different rules at multiple granularities to classify each data-point as normal or abnormal. In [23], a Bayesian network is used to model the underlying distribution. Some initial deep learning-based approaches such as [24], and [25] still rely on designing the input features in the training set. Some years later, more sophisticated approaches such as using a fully automated LSTM auto-encoder are proposed [26, 27]. Approach by Bouritsas *et al.* [26] and Ji *et al.*are applicable even for non-human anomalies, but they are not performing well on large scale anomaly datasets such as Shanghai-Tech [2]. These methods do not include any social interaction for anomaly detection using trajectories.

There exist some research using social interaction for trajectory prediction. Some examples are Gupta *et al.* [7] and Alahi *et al.* [28]. The basic architecture of both approaches includes a single LSTM for each trajectory followed by a social pooling layer to model the interaction. Social GAN [7] however encouraged diverse prediction by including variety loss, which leads to the prediction of near to real trajectories. In this paper, we propose a novel method for prediction-based anomaly detection using trajectories. Our architecture is mainly motivated by Social GANs [7], where we classify the socially possible trajectories to normal or abnormal based on their prediction error. We then show that this prediction-based anomaly detection system utilizing temporal information in form of trajectories can complement spatial-based anomaly detection sytems, resulting in SoTA performance on two benchmark datasets. To the best of our knowledge, none of the previous works for anomaly detection on surveillance datasets explored the inclusion of socially acceptable trajectories generated via tracker as an additional cue.

3 Proposed System

The block diagram of the proposed system is shown in Figure E.1. The main idea of our proposed approach is to utilize social interaction embedded in trajectories to develop a temporal-based anomaly detection system and then use that to complement SoTA spatial-based anomaly detection systems. To achieve this, our proposed system contains two branches, i.e., the spatial branch, which detects the anomalies by mostly using image features, and the temporal branch, which detects the anomalies using trajectories.

In this paper, we use two different SoTA methods for our spatial branch, i.e., Liu *et al.* [2] and Park *et al.* [3], which produce spatial anomaly detection scores. Section 3.2 contains a detailed description of the spatial baseline methods used in our approach. The input to the temporal branch are trajectories, obtained by running tracker [29] on CUHK Avenue [8] and Shanghai-Tech [2]. The generated trajectories are provided as input to the prediction-based anomaly detection network, which also incorporates the features involved with social interaction among the different trajectories. The proposed prediction-based anomaly detection is based on Social GANs [7] and is described in section 3.1. Once we have anomaly score estimated from both spatial and temporal branches, a weighted score-level fusion is performed to generate the final scores.

3.1 Temporal Branch

We propose a method based on Social GAN [7] to detect anomalous trajectories. The generator ($\mathcal{G}_{SGAN}$) network is an LSTM based encoder-decoder, where one LSTM is used for predicting a single trajectory. The prediction of human trajectories in a crowded scene also depends on social interaction among different human beings. Therefore, $\mathcal{G}_{SGAN}$ contains a social pooling module to encode this interaction. The discriminator ($\mathcal{D}_{SGAN}$) is a LSTM encoder network that classifies the output trajectories as real or fake and encourages the generator to predict socially possible trajectories.

The input to the generator ($\mathcal{G}_{SGAN}$) network is a fixed number of past tracklets from the generated trajectories, which in turn further generates a fixed number of future tracklets. Attention-gated tracker [29] is used to generate the trajectories on CUHK Avenue [8] and Shanghai-Tech [2] datasets. The objective function used for predicting future trajectory is the combination of average displacement error (ADE), final displacement error (FDE), and variety loss. ADE is computed as l_2 distance between the predicted and actual points in the future trajectory, FDE is the deviation in the final position with respect to ground-truth (GT), and variety loss is added to mitigate the redundancy in the predicted trajectories. To transform the trajectory prediction network for anomaly detection, i.e., detecting socially unacceptable trajectories, we compute the total error (TE) by combining ADE and FDE for each tracklet. The tracklet is finally classified as normal or anomalous based on the Total Error (TE), which is also called here as temporal anomaly detection score:

$$TE(t) = ADE(t) + FDE(t), \tag{E.1}$$

$$S_{sgan}(t) = \frac{TE(T_t, \hat{T}_t) - \min_t TE(T_t, \hat{T}_t)}{\max_t TE(T_t, \hat{T}_t) - \min_t TE(T_t, \hat{T}_t)}, \tag{E.2}$$

where, T and $\hat{T}$ are actual and predicted trajectory, respectively, and $S_{sgan}(t)$ is the normalized score obtained from social GANs for each tracklet t. We later combine the normalcy scores from temporal and spatial branches. Therefore, we update the total error ($S_{sgan}(t)$) obtained from social GAN (Equation E.2) to obtain the normalcy score ($S_{temporal}$), which is also called the temporal network output in this work:

$$S_{temporal}(t) = (1 - S_{sgan}(t)), \tag{E.3}$$

3.2 Spatial Branch

To show that the proposed temporal-based anomaly detection system using trajectories can improve the performance of different spatial-based anomaly detection systems, we use two different networks in different experiments in the spatial branch of our proposed system. These are future frame prediction-based by Liu *et al.* [2] and memory-based reconstruction/prediction by Park *et al.* [3]. The prediction-based by Liu *et al.* [2] proposed a GAN based method, where generator network aims to generate realistic future frames and discriminator module aims to discriminate between real and generated future frames. Finally, the generated future frame is classified as abnormal or normal based on its quality. The generated normal frames have better quality in comparison to the abnormal frames. This network uses minimal temporal information in the form of optical flow between consecutive frames and optimizes for intensity, gradient, and flow loss. The memory-augmented anomaly detection by Park *et al.* [3] contains an additional memory module which records prototypical pattern of normal data. The memory module is included with both prediction and reconstruction based anomaly detection networks. Park et al. [3] uses convolutional auto-encoders for both reconstruction and prediction networks. It optimizes both prediction/reconstruction auto-encoders by minimizing prediction/reconstruction, compactness, and separateness loss. The compactness loss encourages the query to the nearest item in the memory and the separateness loss encourages the discriminative power of the memory items. Peak Signal to Noise Ratio (PSNR) by Mathieu *et al.* [30], a commonly used method for image quality assessment, is used for evaluating the predicted/reconstructed frames in both cases:

$$PSNR(I,\hat{I}) = 10\log_{10}\frac{[\max_{\hat{I}}]^2}{\frac{1}{N}\sum_{i=0}^{N}(I_i-\hat{I}_i)^2}, \tag{E.4}$$

where, I is actual and $\hat{I}$ is predicted/reconstructed frame. Higher PSNR of the predicted frame increases the probability of it being normal. Then the PSNR score calculated for each frame in a video to generate the spatial anomaly detection score (E.5) [2]:

$$S_{spatial}(t) = \frac{PSNR(I_t,\hat{I}_t) - \min_t PSNR(I_t,\hat{I}_t)}{\max_t PSNR(I_t,\hat{I}_t) - \min_t PSNR(I_t,\hat{I}_t)}, \tag{E.5}$$

where, $S_{spatial}(t)$ is the normalized score for t_{th} frame, I_t and $\hat{I}_t$ are actual and predicted/reconstructed frame, respectively, for tracklet t. .

3.3 Parameter Learning

We propose a parameter learning approach to fuse the information from spatial branch and temporal branch at the score level. Thus, we learn two parameters, one for each score vector. The fusion is defined as follows:

$$S_{Total}(t) = \mathcal{F}(\alpha S_{spatial}(t) + \beta S_{temporal}(t)), \tag{E.6}$$

where α and β are the parameters that we learn to weigh the contribution of spatial network output ($S_{spatial}$) and temporal network output ($S_{temporal}$), respectively. $\mathcal{F}$ is the activation function which is Sigmoid in our case. To form the learning problem, we minimize the binary cross-entropy loss function.

4 Experiments and Results

This section contains details of the evaluation metric, datasets and implementation used in our experiments. The later part of this section also contains quantitative and qualitative results documenting the performance of the introduced temporal-based anomaly detection system using the socially unacceptable trajectories, and its contribution to the proposed system when used with spatial-based anomaly detection systems.

4.1 Evaluation Metrics

The proposed system is evaluated using Receiver Operation Characteristic (ROC) [31] obtained by changing the normality threshold, i.e., fused scores obtained from spatial and temporal network in our case. Area Under the Curve (AUC) is a cumulative measure of accuracy for all possible normality thresholds and used for the accuracy evaluation. A higher value of AUC indicates a better system.

4.2 Datasets

We used two publicly available datasets namely CUHK Avenue [8] and Shanghai-Tech [2] for the training of the baseline models. CUHK Avenue [8] contains 16 training and 21 testing videos with a total of 47 anomalous events. The anomalous events in this dataset are loitering, running, and throwing objects. Shanghai-Tech [2] contains 330 training and 107 test videos with 130 abnormal events. The anomalous events are snatching, chasing, running, fighting, cyclist and vehicles on pedestrian street.

To train the temporal anomaly detection network, trajectory datasets are generated by providing training and testing images from CUHK Avenue [8] and Shanghai-Tech [2] to the attention-gated tracker of Madan *et al.* [29]. The tracking results contain the coordinates of the bounding box along with the object (Identification) ID. The obtained results are converted to a trajectory dataset by converting bounding box coordinates to the center location. Each center position along with the associated ID represents a single tracklet. Object positions associated with the same ID are joined together to form a single trajectory.

4.3 Training and Testing the Proposed System

The baseline architectures of [2, 3] are trained for 15 epochs each on Nvidia RTX 2080 Ti GPU on Shanghai-Tech [2] dataset, which took $\sim$12hrs to complete. We use pre-trained models for CUHK Avenue [8] dataset. Temporal network is trained for 200 epochs individually for each dataset with a batch size of 64 on Nvidia RTX 2080 Ti GPU, which took $\sim$2hrs to complete the training.

At the testing time, we obtain the score vectors from each spatial and temporal branch of our proposed system, which are provided as input to our paramater learning scheme. The learned parameters are used to weigh the spatial and temporal anomaly scores to generate the final scores. We performed micro-level evaluation, as done in [3, 5], where we concatenate all the sequence and learned the parameters for the entire dataset.

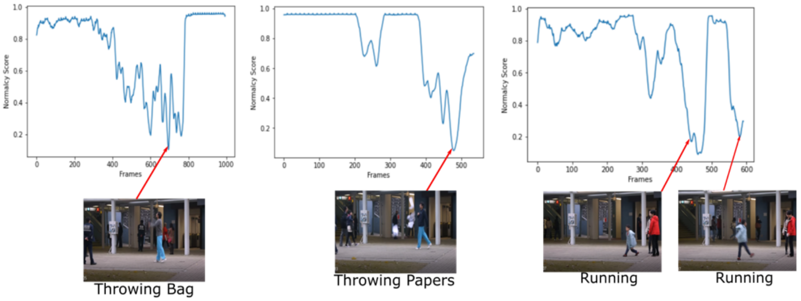


Fig. E.2: Illustrating the anomalies detected by our strategy on avenue dataset. This includes mostly individual anomalies such as throwing bag (left), throwing paper (middle) and running (right).

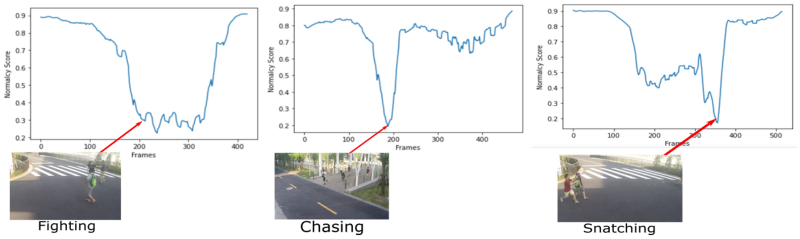


Fig. E.3: Illustrating the anomalies detected by our strategy on Shanghai-Tech [2] dataset, involving social interaction such as fighting (left), chasing (middle), and snatching (right).

4.4 Qualitative results

Figure E.2 and E.3 illustrate visual results on CUHK Avenue [8] and Shanghai-Tech [2] datasets. CUHK Avenue [8] mostly contains individual anomalies, which includes limited social interaction, but our proposed combination still improved the anomaly detection by considering individual trajectories. On the other hand, anomalies in Shanghai-Tech [2] involve small groups interaction such as snatching, fighting. Figure E.3 depicts that the proposed combination detected anomalies like fighting, chasing, and snatching, all of which involve interaction between two people. Thus, our method improves anomaly detection not only in the case of social interaction, but also involving individual trajectories.

As an illustration of a corrected case, Figure E.4 shows an anomaly corresponding to a person moving back and forth to pick-up the bag. This anomaly remains undetected by the baseline method, i.e., Liu *et al.* [2]. However, it is detected by the proposed system. The reason is that continuous back and forth motion is considered as an unacceptable social trajectory.

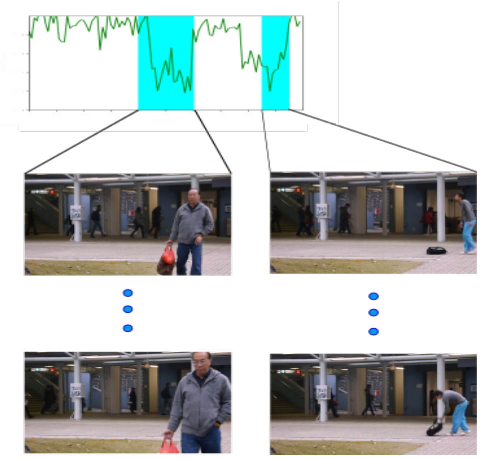


Fig. E.4: An example of an anomaly sequence "walking in wrong direction and throwing bag", from sequence 6 in CUHK Avenue [2], is not detected by the baseline method but it is detected when complemented with trajectory information using the proposed system.

4.5 Quantitative Results

As depicted in Table E.1, the AUC score on CUHK Avenue [8] and Shanghai-Tech [2] using only temporal branch are 65.0% and 69.7%, respectively. It can be observed from these results that trajectories alone are unable to generate competitive results against SoTA methods. The trajectories used in our experiments are constructed using center point, which do not contain much information about the spatial and appearance features of the different objects. Therefore, anomaly detection by simply using these trajectories generate lower AUC scores compared to SoTA. However, when fused with spatial information, as illustrated in Figure E.1, temporal information generated by socially acceptable trajectories contributes in increasing the performance of SoTA spatial-based anomaly detection systems by a large margin, as shown in Table E.1.

It can be observed from the results shown in Table E.1 that the proposed system outperforms listed SoTA approaches including our baseline architecture by Liu *et al.* [2] on CUHK Avenue [8] by 3.4% and on Shanghai-Tech [2] by 1.8%. It outperforms the baseline architecture by Park *et al.* [3] in both forms of 1) reconstruction-based: CUHK Avenue [8] by 4.1% and Shanghai-Tech [2] by 3.3% and 2) prediction-based: CUHK Avenue [8] by 0.1% and Shanghai-Tech [2] by 3.4%. It can be observed from Table E.1 that the information from trajectories is complimenting the baseline architectures irrespective of the underlying network architecture in the spatial branch of

Method	Avenue(%)	SH-Tech(%)
Hasan *et al.* [32]	80.0	60.9
Del *et al.* [33]	78.3	-
Luo *et al.* [34]	77.0	-
Hinami *et al.* [34]	80.9	-
Lu *et al.* [34]	80.9	-
Ionescu *et al.* [35]	80.6	-
Luo *et al.* [35]	81.7	68.0
Liu *et al.* [36]	84.4	-
Ours (Temporal Only: SGAN)	65.0	69.7
Spatial Only: Liu et. al. [2]	85.1	72.8
Ours (Spatial: Liu et. al., Temporal: SGAN)	**86.8**	**74.6**
Spatial Only: Park et. al. - Pred [3]	88.5	70.5
Ours (Spatial: Park et. al. - Pred., Temporal: SGAN)	**88.6**	**73.8**
Spatial Only: Park et. al. - Reconst [3]	82.8	69.8
Ours (Spatial: Park et. al. - Reconst., Temporal: SGAN)	**86.9**	**73.2**

Table E.1: Comparing the frame-level AUC score (in %) of the proposed system with the SoTA approaches and their corresponding spatial anomaly detection branch. Higher frame-level AUC indicate the better performance.

our proposed system. We didn't compare our results against other SoTA approaches, like [10, 16, 18] in this table as they use additional prior knowledge in form of object-detection, which could be included in our system as future work.

Furthermore, the proposed approach does not optimize the feature space with any additional supervision. Some approaches such as Geogescu *et al.* [5] and Feng *et al.* [37] use additional supervision with pseudo labels to improve the latent features, enhancing accuracy of anomaly detection. On the other hand, our approach learns an accurate fusion of temporal and spatial scores without modifying the underlying feature space through additional supervision. Weakly supervised approach by Geogescu *et al.* [5] has an AUC of 92.3% on CUHK Avenue [8] and 82.7% on Shanghai-Tech [2]. Weakly supervised approach of Feng *et al.* [37] has an AUC of 94.3% on Shanghai-Tech [2]. Comparing with weakly supervised approaches, we observed that our approach has competitive results while having less supervision.

Baseline		Proposed	
Dataset	AUC(%)	Train → Test	AUC(%)
CUHK Avenue	85.1	Shanghai-Tech → CUHK Avenue	86.9
Shanghai-Tech	72.4	CUHK Avenue → ShanghaiTech	73.1

Table E.2: Cross-data experiments depicting that the learned parameters on one dataset improves the scores on another.

4.6 Cross-data Evaluation Results

We also verified that learning parameters on a source dataset and testing them on a target dataset with similar anomalies also improves the overall score. We used prediction-based anomaly detection by Liu *et al.*as the baseline for this experiment. It can be observed from Table E.2 that the AUC on Shanghai-Tech [2], i.e., 73.1% is better than the baseline, i.e., 72.4% by 0.7% and CUHK Avenue [8], i.e., 86.9% is better than baseline, i.e., 85.1% by 1.8%.

5 Conclusion

In this paper we hypothesize that temporal information obtained from socially unacceptable trajectories can be used for developing a temporal-based anomaly detection system. Then, we further hypothesize that such a temporal-based anomaly detection system can contribute to improving the performance of SoTA spatial-based anomaly detection systems. To verify these, we propose a system with two branches (one for the spatial and one for the temporal domain) that fuses the results of the two domains at score level. We verify that socially unacceptable trajectories provide discriminative information to identify anomalies in real world surveillance datasets, for two different spatial-based systems employed in the spatial branch of our system. We plan as future work to evaluate different temporal and spatial anomaly detection models in both branches of the proposed scheme and analyze for their complementarity. We also plan to incorporate the prior knowledge from object detection or skeleton for anomaly detection.

Acknowledgements

This work was supported by the Milestone Research Programme at Aalborg University (MRPA), by the Spanish project PID2019-105093GB-I00 (MINECO/FEDER, UE), and by ICREA under the ICREA Academia programme.

References

[1] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 733–742.

[2] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection - a new baseline," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6536–6545.

[3] 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.

[4] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[5] M. I. Georgescu, R. Ionescu, F. S. Khan, M. Popescu, and M. Shah, "A background-agnostic framework with adversarial training for abnormal event detection in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[6] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, "A survey of single-scene video anomaly detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. PP, 2020.

[7] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[8] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[10] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 1577–1581.

[11] T.-N. Nguyen and J. Meunier, “Anomaly detection in video sequence with appearance-motion correspondence,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1273–1283.

[12] R. Rodrigues, N. Bhargava, R. Velmurugan, and S. Chaudhuri, “Multi-timescale trajectory prediction for abnormal human activity detection,” in *The IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[13] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, “Spatio-temporal autoencoder for video anomaly detection,” in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1933–1941.

[14] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, “Learning regularity in skeleton trajectories for anomaly detection in videos,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, “Integrating prediction and reconstruction for anomaly detection,” *Pattern Recognition Letters*, vol. 129, pp. 123–130, 2020.

[16] R. T. Ionescu, F. S. Khan, M.-I. Georgescu, and L. Shao, “Object-centric auto-encoders and dummy anomalies for abnormal event detection in video,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[17] M.-I. Georgescu, A. Barbalau, R. T. Ionescu, F. S. Khan, M. Popescu, and M. Shah, “Anomaly detection in video via self-supervised and multi-task learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 12 742–12 752.

[18] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, “Cloze test helps: Effective video anomaly detection via learning to complete video events,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 583–591.

[19] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, “Efficient anomaly monitoring over moving object trajectory streams,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. New York, NY, USA: Association for Computing Machinery, 2009, p. 159–168. [Online]. Available: https://doi.org/10.1145/1557019.1557043

[20] C. CHEN, D. Zhang, P. S. CASTRO, N. Li, L. Sun, S. LI, and Z. WANG, “iBOAT : isolation-based online anomalous trajectory detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806–818, June 2013. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00831500

[21] Naohiko Suzuki, Kosuke Hirasawa, Kenichi Tanaka, Yoshinori Kobayashi, Yoichi Sato, and Yozo Fujino, "Learning motion patterns and anomaly detection by human trajectory analysis," in *2007 IEEE International Conference on Systems, Man and Cybernetics*, 2007, pp. 498–503.

[22] X. Li, J. Han, S. Kim, and H. Gonzalez, "ROAM: rule- and motif-based anomaly detection in massive moving object data sets," in *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007, pp. 273–284. [Online]. Available: https://doi.org/10.1137/1.9781611972771.25

[23] F. Johansson and G. Falkman, "Detection of vessel anomalies - a bayesian network approach," in *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, 2007, pp. 395–400.

[24] C. Ma, Z. Miao, M. Li, S. Song, and M. Yang, "Detecting anomalous trajectories via recurrent neural networks," in *Computer Vision – ACCV 2018 - 14th Asian Conference on Computer Vision, Revised Selected Papers*, ser. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), G. Mori, H. Li, C. Jawahar, and K. Schindler, Eds. Germany: Springer Verlag, 2019, pp. 370–382, 14th Asian Conference on Computer Vision, ACCV 2018 ; Conference date: 02-12-2018 Through 06-12-2018.

[25] D. Yao, C. Zhang, Z. Zhu, J. Huang, and J. Bi, "Trajectory clustering via deep representation learning," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3880–3887.

[26] G. Bouritsas, S. Daveas, A. Danelakis, and S. C. A. Thomopoulos, "Automated real-time anomaly detection in human trajectories using sequence to sequence networks," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–8.

[27] Y. Ji, L. Wang, W. Wu, H. Shao, and Y. Feng, "A method for lstm-based trajectory modeling and abnormal trajectory detection," *IEEE Access*, vol. 8, pp. 104 063–104 073, 2020.

[28] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[29] N. Madan, K. Nasrollahi, and T. B. Moeslund, "Attention-enabled object detection to improve one-stage tracker," in *Accepted in Intelligent Systems Conference (IntelliSys)*, 2021.

[30] M. Mathieu, C. Couprie, and Y. LeCun, “Deep multi-scale video prediction beyond mean square error,” Jan. 2016, 4th International Conference on Learning Representations, ICLR 2016 ; Conference date: 02-05-2016 Through 04-05-2016.

[31] T. Fawcett, “An introduction to roc analysis.” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Available: http://dblp.uni-trier.de/db/journals/prl/prl27.html#Fawcett06

[32] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, “Learning temporal regularity in video sequences,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[33] A. Del Giorno, J. A. Bagnell, and M. Hebert, “A discriminative framework for anomaly detection in large videos,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 334–349.

[34] C. Lu, J. Shi, and J. Jia, “Abnormal event detection at 150 fps in matlab,” in *2013 IEEE International Conference on Computer Vision*, 2013, pp. 2720–2727.

[35] W. Luo, W. Liu, and S. Gao, “A revisit of sparse coding based anomaly detection in stacked rnn framework,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[36] Y. Liu, C. Li, and B. Póczos, “Classifier two sample test for video anomaly detections,” in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018.* BMVA Press, 2018, p. 71. [Online]. Available: http://bmvc2018.org/contents/papers/0237.pdf

[37] J.-C. Feng, F.-T. Hong, and W.-S. Zheng, “Mist: Multiple instance self-training framework for video anomaly detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 14 009–14 018.

Paper F

CM-MAE: Curriculum Masking for learning representation in Mask Autoencoders

Neelu Madan, Nicolae-Cătălin Ristea, Radu Tudor Ionescu, Kamal Nasrollahi, Thomas B. Moeslund

This is a 'Technical Report' covering work in progress
that hasn't been submitted to a conference

Abstract

Masked image modeling has been demonstrated as a powerful pretext task for generating robust representations that can be effectively generalized across multiple tasks. Typically, this approach involves randomly masking tokens in the image, with the masking strategy remaining consistent throughout training. In this paper, we propose a curriculum masking approach that updates the mask to continually increase the complexity of the pretext task. The underlying idea is that by gradually increasing the task complexity, the model can learn more robust and transferable representations. To facilitate this, we introduce a novel masking module that possesses the capability to generate masks of different complexities. In our current implementation, we update the curriculum at fixed intervals, and the zero-shot results achieved with this approach are comparable with the baseline method (MAE). We trained and evaluated the model on 10% of the ImageNet data in the proposed setup and observed that our initial results are comparable to the baseline. In these settings, we proposed to train both the baseline and masking modules by the same amount. As the gradient of the masking module unstabilizes during the adversarial training, we propose frequent updates to our curriculum. We finally propose a configuration where we stop training the masking module early and extend the interval of curriculum update. This configuration provided us with the initial results, where our curriculum masking-MAE (CM-MAE) outperforms the baseline on Acc@1 by 2.9% and Acc@5 by 3.5%. As these are only initial results, we plan to further extend this work on large-scale training on ImageNet-1K and transfer the representation to the different visual tasks.

1 Introduction

Self-supervised representation learning aims at learning the representation is become a prominent research topic as the learned representation can be generalized over multiple visual tasks ranging such as image recognition [1–3], object detection [4–6] and segmentation [7–10]. These representations are learned based on defining a self-supervised task also known as *pretext task*, where the labels are defined based on the available data. Masked language modeling [11] techniques in natural language processing (NLP), the field of computer vision has embraced masked-image-modeling [12–16, 16–22] as a self-supervised task. Masked image modeling involves masking a portion of an image and the model in turn learns to reconstruct the masked information based on the remaining unmasked region.

The existing literature on using MIM as a self-supervised learning technique can be categorized into two main groups, based on the reconstruction target: visual tokens [12, 13, 15, 15, 16, 16, 18, 18, 23, 24, 24] and features [17, 25]. Among these, predicting masked tokens has emerged as the most prominent approach, primarily due to its simplicity and better generalization capabilities. While previous research has dedicated considerable attention to refining the pretext-task [12, 15, 21, 24, 26–30], comparatively less emphasis has been placed on token selection strategies [16, 22, 31, 32]. Some of the existing studies base their token selection on semantic object parts [16, 22], adversarial masking [21], attention-guided techniques [33], and windowed masking [31]. In contrast to other approaches where the mask portion is always fixed during the training, we propose a *novel masking module*, which is trained in an end-to-end fashion with an MAE backbone, and poses the capabilities to adapt masks, in order to increase the complexity of the pretext task, during the training process.

Our proposed method introduces curriculum learning as the fundamental concept, and uses MAE [24] as the underlying backbone for representational learning. Curriculum learning, originally introduced by Bengio et al. [34], operates on the principle that models learn to solve tasks in a progressive manner, starting from simpler tasks and gradually advancing to more complex ones. This approach facilitates the acquisition of robust representations and enhances generalization capabilities. To implement curriculum learning, we propose a *novel masking module* that generates masks of varying complexities. The complexity of the generated masks is controlled by the progressive loss factor within our proposed masking module. We train the proposed masking module in two stages: 1) The masking module facilitates the MAE backbone and generates masks of relatively lower complexities, and 2) The masking module is trained in an adversarial manner with the MAE backbone, gradually increasing the complexity of the output masks. The complexity of the output mask in this case is increasing by varying the adversarial weight to in the training process of our masking module. The curriculum is thus updated based on the mask generated with different complexity levels.

In our experiments, we pre-train our model using a subset of the ImageNet-1K dataset, specifically consisting of 10% of the classes. We evaluate the performance of our models in zero-shot settings on 100 classes from the ImageNet dataset. In our preliminary approach, we obtain Acc@1 as 38.0 and Acc@5 as 60.4, which are almost comparable to the baseline method, *i.e.*, MAE [24]. This method considers a frequent curriculum update strategy, aiming to address the issue of unstable gradients arising from longer adversarial training. Subsequently, we make adjustments to our settings by increasing the interval at which the curriculum is updated. This modification mitigates the problem of unstable gradients during training. Specifically, we train the proposed masking module for only 10% of the total training time during each curriculum update. By adopting this revised approach, we alleviate the instability of gradients and foster more stable and effective training of the masking module.

In our revised experiment settings, we observe a noteworthy improvement in performance. Specifically, we achieve a substantial increase of 2.9% in the top-1 accuracy (Acc@1) and a notable improvement of 3.6% in the top-5 accuracy (Acc@5). These promising results encourage us to pursue further investigations in this specific experimental setup. We plan to expand our experiments by incorporating additional visual tasks. Additionally, we also plan to include curriculum masking (CM) with different underlying backbones, such as ViT-B, ViT-L, and ViT-H, under various settings. Overall, the main contribution of our work can be summarized below:

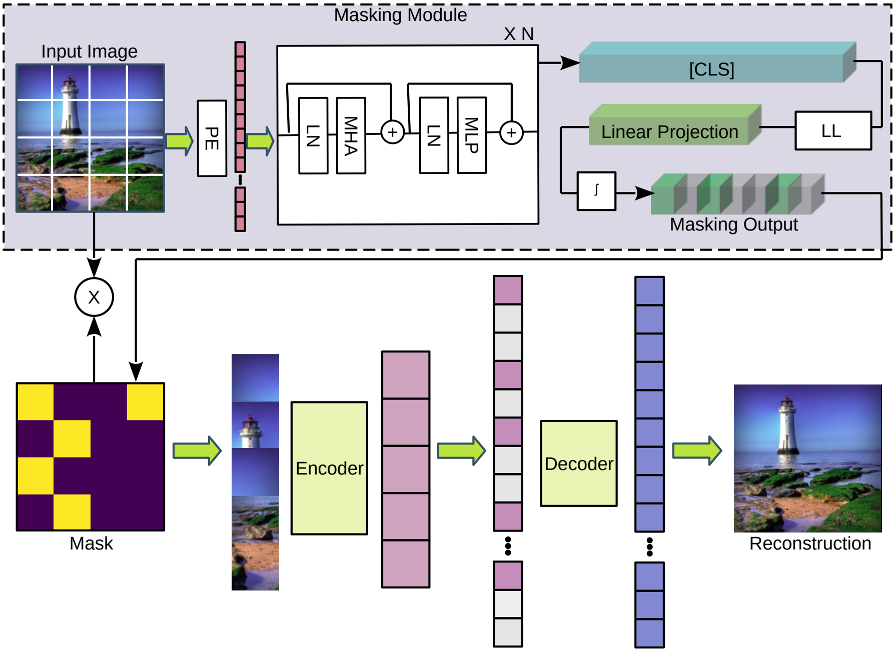


Fig. F.1: The overall architecture to enable curriculum learning while using MAE as the backbone for self-supervised representation learning. The masking module generated the masking output based on the complexity of the task and provide the visible tokens to MAE based on this output.[Best Viewed in Color]

- We proposed a curriculum learning in a classical setting, i.e., easy-to-hard in the MAE framework to learn robust representation.
- We proposed a novel learnable masking module, which possesses the capability to generate adaptive masks based on task complexity.
- We present initial experimental settings and results for our proposed curriculum masking approach, demonstrating its potential to outperform existing methods and support our hypothesis.

2 Related Work

Self-supervised Representation Learning. Among the state-of-the-art methods in self-supervised learning, two prominent categories are using discriminative and generative tasks [12–20, 22, 23, 35–39]. Discriminative tasks, like contrastive learning (CL) [35–39], aim to learn robust representations by bringing similar pairs closer and pushing dissimilar pairs apart. Approaches differ in how they handle negative image pairs, with SimCLR [36] employing one-to-one comparisons, MOCO [35] utilizing a momentum encoder for a dynamic negative sample dictionary, and BYOL [37] relying solely on positive pairs. In contrast, generative tasks, specifically masked image modeling (MIM) [12–16, 16–20, 22, 23], focus on reconstructing masked image regions using visible context. MIM simplifies the pretext task by randomly masking a significant portion of an image, which He *et al.* [24] demonstrate as a challenging yet effective approach for learning robust representations without heavy data augmentation. Due to its simplicity and generalizability, we incorporate MIM as our chosen pretext task.

Masked Image Modeling(MIM). A large amount of research nowadays in self-supervised representation learning is based on MIM [12–16, 16–20, 22, 23], which is highly inspired from BERT [11] using masked language modeling (MLM). The different approaches based on the reconstruction target in this domain can be categorized as visual tokens [12, 13, 15, 16, 18, 23, 24], and features [17, 25]. The initial approaches centered around predicting visual tokens typically rely on an external tokenizer, which creates a visual codebook corresponding to the reconstruction targets. BeiT [12] and PeCo [13] is based on generating an offline visual codebook using variation autoencoder following DaLL-E [40], iBOT [23] later proposed an online tokenizer based on teacher-networks generated via self-distillation process. To mitigate the requirement of generating a visual codebook, Wei *et al.* [17] proposed to reconstruct the HOG features for the masked region. These approaches are now replaced with relatively simpler methods [14–16, 18, 21, 22, 31] that use pixel values directly as the reconstruction target. MAE [24] proposes aggressive masking of 75% of the image patches randomly resulting in an optimal pretext task to learn robust and generic representation. SimMIM [15] increases the complexity of the pretext task by increasing the patch size and reducing the decoder network to a single layer and claims that a harder pretext task leads to a better representation.

Within the subcategory of MIM approaches that focus on reconstructing masked tokens, various masking strategies have been proposed to obtain robust representations [16, 21, 22, 31]. Li et al. [31] introduce a token selection strategy specifically designed for Pyramid-based Vision Transformer (ViT), as random selection does not yield satisfactory results in this context. Kakogeorgiou et al. [33] propose a student-teacher training framework, where the teacher model learns to identify suitable masking tokens to facilitate representation learning for the student model. SemMAE [16] suggests semantically guided masking, utilizing two modules: a self-supervised part generator and the MAE [24] for representation learning. Building upon SemMAE, AutoMAE [22] introduces a unified framework that integrates the part generator and MAE into a single differentiable architecture. In contrast to these existing approaches, we propose an adaptive masking strategy throughout the training process. Our approach takes into consideration the varying complexity of tasks over time, enabling the generation of masks that align with the specific task complexity at any given stage. To facilitate this adaptive masking, we introduce a *novel masking module* with the capability to adaptively generate masks corresponding to different levels of complexity

Curriculum Learning. Curriculum learning, as introduced by Bengio et al. [34], is a strategy aimed at organizing input data or tasks in a meaningful order to enhance the overall learning outcome. It consists of two main components: curriculum criteria [34] and scheduling function [34]. Approaches in curriculum learning can be categorized into easy-to-hard and hard-to-easy paradigms. In the easy-to-hard paradigm, tasks are presented to the model in increasing order of complexity [32, 34, 41–43]. Conversely, the hard-to-easy paradigm reverses the order [44, 45]. Constructing a curriculum involves either a manual approach using visible complexity measures, such as occlusion degree and shape complexity [34, 46], or employing self-paced learning techniques [47–50], where a neural network dynamically assesses the difficulty of training samples based on their loss. The scheduling function determines when to update the training process and can be categorized as discrete or continuous. In the discrete scheduler [34, 51], data is sorted and divided into discrete sets according to the curriculum criteria. In the case of a continuous scheduler [49, 52], the model is provided with a gradually increasing proportion of difficult training samples, ranging linearly from 0 to 1, where 0 represents no hard samples and 1 represents all hard samples. We proposed a method to incorporate easy-to-hard scheduling based on the complexity of the pretext task in the proposed approach, where the model's output, i.e., reconstruction error is used as a measure to construct such curricula.

In this paper, we propose a curriculum learning approach where we progressively increase the complexity of the pretext task from easy to hard. Our approach utilizes a *novel masking module* that generates masks of varying complexity, corresponding to different difficulty levels. We employ a discrete scheduler to facilitate curriculum updates, ensuring that samples within each update share the same difficulty level. This structured approach enables systematic learning and the development of robust representations.

3 Method

Our proposed approach incorporates curriculum learning to introduce challenging tasks to the MAE [24] backbone, enabling the model to learn and solve them effectively. In this section, we provide an overview of our methodology, including the ***Curriculum Masking*** setup, the ***Masking Module*** for generating flexible masks, and the integration of different ***loss functions*** to promote curriculum learning.

3.1 Curriculum Masking

We incorporate curriculum learning in our approach to gradually increase the difficulty of the pretext task and the complexity of the generated masks. To accomplish this, we propose a novel masking module that dynamically masks tokens, providing challenging tasks for the MAE backbone [24]. The training of the masking module is conducted in two stages. In the first stage, the module is trained to make the pretext task easier by aligning its gradients with the MAE backbone. In the second stage, we reverse the gradients of the masking module with respect to the MAE, facilitating adversarial learning between the two frameworks. During adversarial training, the masking module selects hard tokens that impede reconstruction, thereby increasing the difficulty of the pretext task. We advocate updating the curriculum at fixed intervals of epochs (η), gradually enhancing the complexity of the generated masks. This complexity increment is achieved by adjusting the progressive loss factor ($\mathcal{L}_{prog}$), thereby amplifying the adversarial training.

3.2 Masking Module

The curriculum masking module (CMM) takes an input image $\boldsymbol{I} \in \mathbb{R}^{h \times w \times c}$, the module produces a masking vector $\boldsymbol{Z} = \{z_i\}_{i=1}^{n}, z_i \in \{0, 1\}$ as output, where n represents the number of patches. This vector indicates which patches should be masked. We further constrain the masking output using the Gaussian loss ($\mathcal{L}_{GL}$) to converge towards either zero or one. The top-K tokens from the masking output are selected as the visible tokens ($\boldsymbol{V}$), where the value of K is determined based on a fixed masking ratio.

The CMM operates by dividing the input image $\boldsymbol{I}$ into $n = hw/p^2$ non-overlapping patches, each of size $p \times p$. These patches are then projected using a linear layer to generate patch embeddings, which are combined with positional encoding to obtain embedded tokens $\boldsymbol{T} = \{t_i\}_{i=1}^{n}, t_i \in \mathbb{R}^d$, where d is the token dimension. The embedded tokens, along with a learnable $[CLS]$ token, are fed into a transformer-based backbone inspired by the architecture of Vision Transformer (ViT) [1]. The $[CLS]$ token captures information about the entire input image, and the masking output ($\boldsymbol{Z}$) is generated by mapping the output class token to the number of input patches using a linear layer followed by a sigmoid activation. Various loss functions are employed to facilitate the effective functioning of the masking module, as discussed in the subsequent subsection.

3.3 Loss Functions

In our proposed CMM, multiple loss functions are utilized to support the proposed curriculum learning framework and enhance the training process. These loss functions contribute to the overall objective of generating adaptive masks and training the masking module effectively.

Progressive Loss. We utilize the mean squared error (MSE) metric as our progressive loss function, similar to MAE [24], to measure the discrepancy between the normalized per-patch pixels of the reconstructed target ($\hat{\boldsymbol{I}}$) and the input image ($\boldsymbol{I}$). This progressive curriculum loss function, denoted as $\mathcal{L}_{prog}$, is expressed by Equation F.1.

$$\mathcal{L}_{\text{prog}}(\hat{\boldsymbol{I}}, \boldsymbol{I}) = \begin{cases} (\hat{\boldsymbol{I}} - \boldsymbol{I})^2, & \text{if } Epochs < \eta \\ -(\hat{\boldsymbol{I}} - \boldsymbol{I})^2, & \text{otherwise} \end{cases} \tag{F.1}$$

As mentioned in Section 3, the training of the masking module is conducted in two stages. In the first stage, which lasts for less than η epochs, the loss is positive, encouraging the masking module to produce easy masks. Easy masks refer to masking fewer tokens from the foreground and more tokens from the background. The second stage begins after η epochs, where the loss function is reversed to negative. In this stage, adversarial training is initiated, and the masking module learns to generate hard masks. Hard masks, on the other hand, involve masking more tokens from the foreground and fewer tokens from the background.

Gaussian Loss. We incorporate a Gaussian objective into our proposed masking module to enforce binary outputs. Specifically, the masking module results in 1 if the output patch needs to be masked, and 0 otherwise. To achieve this, we employ the Gaussian loss, which is defined by Equation F.2. In this equation, μ denotes the mean and σ corresponds to the standard deviation and $\boldsymbol{Z}$ corresponds to the masking output. For our implementation, we set the values of μ and σ to 0.5 and 0.12 respectively.

$$\mathcal{L}_{Gauss} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\boldsymbol{Z}-\mu)^2}{2\sigma^2}\right) \tag{F.2}$$

By training the proposed masking module solely with the Gaussian Loss, it acquires the capability to determine whether a specific token should be masked or left unmasked. In our curriculum learning setup, we observed that for the simpler task, the model tends to get stuck in the local minima where it is masking fewer tokens than the predetermined masking ratio as shown in Figure F.5. Conversely, for difficult tasks, the model tends to mask all the tokens as shown in Figure F.4. To address this behavior, our subsequent loss function aims to ensure a consistent number of tokens are masked, irrespective of the task complexity.

Kullback-Leibler (KL) Divergence Loss. The KL-divergence loss ensures that a fixed number of tokens are masked based on a predefined masking ratio, irrespective of the pretext task's complexity. This KL-divergence loss shown by Equation F.3 is included in the objective of our proposed masking module.

$$\mathcal{L}_{KL} = \boldsymbol{M} \cdot \log\left(\frac{\hat{\boldsymbol{M}}}{\boldsymbol{M}}\right) + \boldsymbol{V} \cdot \log\left(\frac{\hat{\boldsymbol{V}}}{\boldsymbol{V}}\right) \tag{F.3}$$

Where, $\boldsymbol{M}$, $\hat{\boldsymbol{M}}$ denotes the actual and output masked tokens, $\boldsymbol{V}$ and $\hat{\boldsymbol{V}}$ represents the actual and output of visible tokens. The fractions $\frac{\hat{\boldsymbol{M}}}{\boldsymbol{M}}$ in the KL Loss equation provide insights into the match between the predicted outputs and the actual values for the masked tokens. Whereas the fraction $\frac{\hat{\boldsymbol{V}}}{\boldsymbol{V}}$ provides insights into the match between the predicted outputs and the actual values for the visible tokens. When the predicted outputs align with the actual values, these fractions evaluate to zero. The scale factors $\boldsymbol{M}$ and $\boldsymbol{V}$ ensure that each fraction is appropriately weighted in the computation of the KL Loss.

Diversity Loss. We further include a diversity loss in order to obtain different masking configurations for each sample. The diversity loss to obtain the different masking

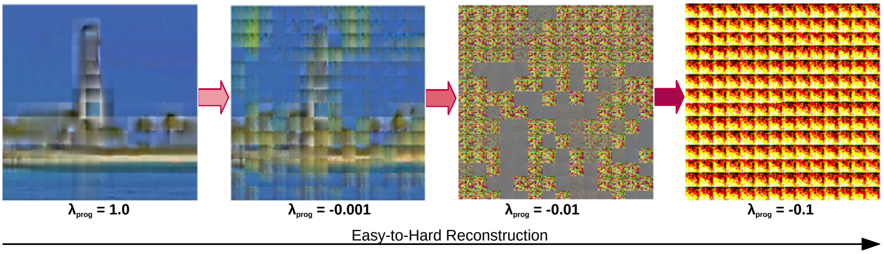


Fig. F.2: Figure shows the reconstruction after the first epoch from easy-hard masking. Left most image show when we are training the masking module to help MAE in the PreText task. The right 3 images show the reconstruction after different adversarial factors.

results is computed using equation F.4.

$$\mathcal{L}_d = \frac{1}{N \cdot (N-1)/2} \sum_{i=1}^{N} \sum_{j=1, j \neq i}^{N} \exp^{-k \cdot D_{ij}} \tag{F.4}$$

Where N represents the number of samples in the minibatch, and D_{ij} is the Euclidean distance between two samples, indexed as i and j, and k is the scaling factor that adjusts the influence of distance on the loss function. The loss function is normalized by the factor of $N(N-1)/2$ in order to make it independent of the batch size.

Total Loss. The total loss function is shown in Equation F.5. In this equation, the hyper-parameters λ_{GL}, λ_{prog}, λ_{KL}, and λ_{div} dictate the contributions of each loss term to the overall loss function.

$$\mathcal{L}_{Total} = \lambda_{prog}\mathcal{L}_{prog} + \lambda_{Gauss}\mathcal{L}_{Gauss} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{div}\mathcal{L}_{div} \tag{F.5}$$

In the framework of our proposed curriculum learning approach, the progressive loss factor (λ_{prog}) plays a pivotal role in shaping the difficulty of the pretext task. As described above the two-stage training of the masking module. In the first stage, $\mathcal{L}_{prog}$ is positive and we set this λ_{prog} to 1 to select easy tokens. In this stage, the masking module is trained for a fixed number of epochs (η) under the simplified pretext task. In the second stage, $\mathcal{L}_{prog}$ is negative, which means the masking module now learns to select hard tokens increasing the complexity of the pretext task. At this stage, we keep increasing the complexity of the pretext task by multiplying λ_{prog} with a factor of 2 after every η epochs. Thus, it sets up a curriculum, where the complexity of the pretext task increases progressively at every fixed number of epochs (η).

By implementing this curriculum learning strategy, we effectively modulate the task's difficulty, providing the model with a learning trajectory that gradually advances from simpler to more challenging representations. This gradual increase in difficulty fosters the model's learning and adaptation capabilities over time, enabling the acquisition of more robust and transferable representations.

4 Experiments and Results

We evaluate our curriculum masking approach in a zero-shot setup on 10% of the ImageNet. For evaluation purposes, we use the ViT base architecture with the input image of 224 × 224 and patch size of 16 × 16. We compare the results against MAE [24] as our baseline. The preliminary results indicate that curriculum masking shows comparable results to the baseline method, i.e., MAE [24]. In our last experiment, we update the configuration of curriculum masking and outperform the baseline by a significant margin.

Significance of progressive-loss factor (λ_{prog}). In our proposed method, we choose a significantly low value of progressive-loss factors based on our experiments. This choice is made with the intention of facilitating curriculum learning. Figure F.2 shows the reconstructed output after the first epoch of the progressive training by varying the values of the progressive loss factor. The factor value of 1.0 serves as an indication that the masking module is effectively learning to simplify the pretext task. To introduce a greater level of complexity, we gradually introduced adversarial training with varying weight values such as -0.001, -0.002, and -0.1. The figures presented in our research demonstrate that the weight factor of -0.1 excessively amplifies the task's complexity, thereby impeding the model's ability to acquire a robust and transferrable representation space. Consequently, we opted to commence the adversarial training process with a low initial value of -0.001 and increase the complexity by a factor of 2 after every η epochs (a fixed number).

Training Masking Module. In order to determine the training strategy for the proposed masking module, we perform two experiments first determining the training configuration and the second ensuring the mask variety.

Table F.1 compares the results when the MAE [24] backbone and masking module are trained in the same cycle and alternate cycles. The results indicate that training them in the same cycle generates better results. Therefore, we continue the same settings across all our experiments.

In order to ensure that our curriculum masking module chooses different masks for each difficulty level, we perform experiments by randomly re-initializing the masking module's parameters before increasing the weight of the adversarial loss. The zero-shot results comparing the performance with and without re-initializing the masking module's parameters are shown in Table F.2. The results indicate that incremental training of the masking module instead of using parameter re-initialization techniques provide better results.

Curriculum Updates. In the context of this paper, we conduct initial experiments to determine the appropriate interval for updating the curriculum. The adversarial training in the second stage of training our masking module results in unstabilizing the training process. In order to keep training both the curriculum masking module and the MAE backbone [24]. We make frequent updates in the curriculum with values of η as 30, and 50 epochs. The experimental results in these configurations are shown

Method	Zero-shot	
	Acc@1	Acc@5
MAE (Baseline) [24]	39.2	61.5
CM-MAE (Same Cycle)	38.1	60.4
CM-MAE (Alternate Cycles)	37.0	59.6

Table F.1: The results compare the baseline with CM-MAE when our masking module and MAE train in the same cycle and in alternate cycles.

Method	Zero-shot	
	Acc@1	Acc@5
MAE (Baseline) [24]	39.2	61.5
CM-MAE (w/o re-initialized parameters)	38.1	60.4
CM-MAE (w/ re-initialized parameters)	37.6	59.6

Table F.2: The results comparing without and with parameter re-initialization of the masking module after each curriculum update.

in Table F.3. The results show that increasing the update interval (η) improves the performance, but the overall results are still comparable to the baselines.

Method	Zero-shot	
	Acc@1	Acc@5
MAE (Baseline) [24]	39.2	61.5
CM-MAE (Curriculum Update - 30 epochs)	37.2	58.9
CM-MAE (Curriculum Update - 50 epochs)	**38.1**	**60.4**

Table F.3: Results after updating the curriculum after $\eta = 30$ epochs and $\eta = 50$ epcohs. The results show that increasing the update interval improves the results.

Significance of diversity Loss ($\mathcal{L}_{div}$). Figure F.4 shows the results before and after introducing the diversity loss $\mathcal{L}_{div}$ into the objective of our masking module. The figure shows that the masking module is trained without adding the diversity loss is end up producing an almost similar mask irrespective of the input sample. However, using $\mathcal{L}_{div}$ ensures a different mask for each sample in a training batch.

Significance of KL Loss ($\mathcal{L}_{KL}$). Figure F.5 and F.4 show the masking outputs for the easy and hard tasks respectively, when KL loss is not introduced into the objective. Figure F.5 indicates that the networks are refraining from masking tokens for the first few epochs in order to perform well on the pretext task (reconstruction). For a masking ratio of 75%, the masking module only be able to mask less than 50% on the image by finding a local minimum. On the other hand, Figure F.4 shows that the model with adversarial training ends up masking all the tokens in order to make the task hard. Therefore, The KL loss that ensures the number of mask tokens complies with the mask ratio irrespective of the complexity of the task.

CM-MAE outperforming the baseline. The experimental results presented in Table F.3 indicate that increasing the curriculum update interval yields better outcomes.

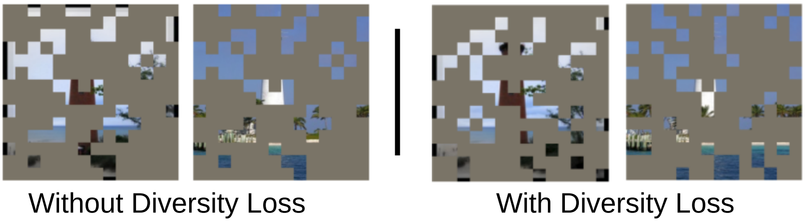


Fig. F.3: Figure shows the generated masks from the proposed masking module without (left) and with (Right) adding the diversity loss ($\mathcal{L}_{div}$). The masking module produces the same masks if diversity loss ($\mathcal{L}_{div}$) is not included in the objective function.

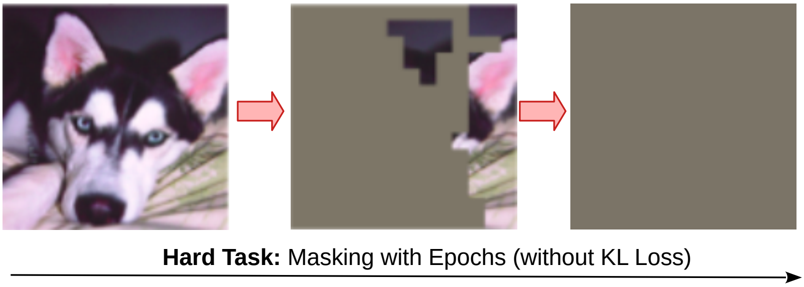


Fig. F.4: Figure shows the results from the masking module on a hard task without including the KL loss. The left image is the input and the middle is the masked output at an intermediate epoch, finally the right image shows the mask at convergence. The model is masking all the tokes to increase the complexity.

Expanding on this finding, we propose a solution where we augment the update interval while restricting the training of the masking module to the initial 20% of epochs within the total interval. This approach effectively addresses the issue of unstable gradients, allowing the model sufficient time to learn the complexities of the pretext task. To evaluate the effectiveness of our curriculum masking approach, we compare the results against a baseline in Table F.4. The findings demonstrate that our proposed approach outperforms the baseline under these conditions. Moving forward, we plan to extend our experiments to various visual tasks and ViT-backbones within this framework, with the expectation of observing improvements across all scenarios.

Method	Zero-shot	
	Acc@1	Acc@5
MAE (Baseline) [24]	39.2	61.5
CM-MAE (Updated Settings)	**42.1**	**65.1**

Table F.4: The preliminary results show our curriculum-masking approach improves over the baseline method. The results show that the proposed approach outperforms the baseline by a significant margin.

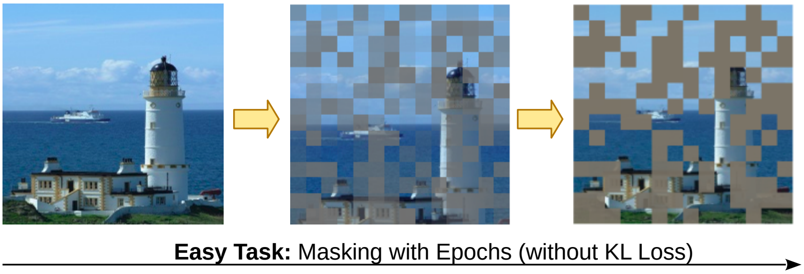


Fig. F.5: Figure shows the results from the masking module on an easy task without including the KL loss. The left image is the input and the middle is the masked output at an intermediate epoch, finally the right image shows the mask at convergence. The model is stuck in local minima and masking less than 75% (predefined masking ratio) tokens.

4.1 Ablation Studies

As the proposed masking module contains multiple loss functions weighted by the hyperparameters λ_{GL}, λ_{KL}, and λ_{div}. Out of which, λ_{prog} is the most crucial factor, which helps in deciding the difficulty of the task. We already discuss the experiments for choosing the appropriate value of λ_{prog} in our main experiments. This section discusses the experiments involved with choosing an appropriate value for λ_{GL}, λ_{KL}, and λ_{div}. The experiments for choosing the appropriate value of the hyperparameters are conducted on 10% on the ImageNet similar to our main experiments. The ablation studies contain the experiments with the curriculum update interval, η, as 50 epochs as we find the new experimental setups later in this experiments.

Tuning the hyper-parameter λ_{GL} and λ_{KL} . Table F.5 presents the results of tuning the hyperparameters λ_{GL} and λ_{KL}. These hyperparameters respectively weigh the Gaussian loss ($\mathcal{L}GL$) and the KL-divergence loss ($\mathcal{L}KL$) in our main objective function. The experiments involved selecting a fixed value of λ_{prog} as 1.0, which allowed training on the easy task for 200 epochs to determine these hyperparameters. The results indicate that setting both λ_{GL} and λ_{KL} to 10 produces the best outcomes, leading us to utilize these values in our primary experiments. Interestingly, we observed that with weighting factors of 10 for both λ_{GL} and λ_{KL}, the magnitudes of the Progressive loss, Gaussian loss, and KL loss become nearly equal. Additionally, a comparison of the results in Table F.5 reveals that including the KL-divergence loss leads to improved performance on the downstream task. This improvement suggests that the network may be masking fewer tokens than the predefined masking ratio (75%), as depicted in Figure F.5, which adversely affects the learned representation and subsequently results in poor transferability to the downstream task.

Tuning the hyper-parameter λ_{div} . The insights gained from the previous ablation study motivated us to select the value of λ_{div} as 2. However, we also conducted additional experiments using different values of λ_{div}. Table F.6 presents the results of these experiments, demonstrating improvements when setting λ_{div} to 5. We use the

Method	$\mathcal{L}_{GL}$	$\mathcal{L}_{KL}$	λ_{GL}	λ_{KL}	Zero-shot	
					Acc@1	Acc@5
$CM - MAE$ (w/ Easy Task)	✓	✗	1	-	35	56.7
	✓	✗	2	-	37.8	59.6
	✓	✗	5	-	35.1	56.7
	✓	✗	10	-	**38.7**	**61.7**
	✓	✓	10	1	36.5	59.3
	✓	✓	10	2	37.8	59.9
	✓	✓	10	5	38.2	60.6
	✓	✓	10	10	**39.5**	**61.9**

Table F.5: Ablation results to choose the values of hyperparameters λ_{KL} and λ_{GL}. Showing the zero-shot classification accuracy on 10% ImageNet, trained for 200 epochs with an easy task.

same values for all our experiments.

Method	Scale	Zero-shot	
		Acc@1	Acc@5
CM-MAE (w/ Curriculum Masking)	1	38.1	60.3
	2	38.1	60.4
	5	38.4	60.4
	10	37.5	59.8

Table F.6: Ablation results to choose the values of hyperparameters λ_{div}. Showing the zero-shot classification accuracy on 10% ImageNet, trained for 200 epochs with the proposed curriculum masking.

5 Discussion

In the context of our preliminary experiments, we have implemented a curriculum update strategy at regular intervals, regardless of the task complexity. Our observations suggest that more complex tasks may require a longer time to converge compared to easier tasks. However, in our current setup, where both the masking module and MAE [24] are trained in the same cycle for improved results (as verified in Table F.1), we encountered stability issues during adversarial training of the masking module. Despite introducing gradient clippings, the training process became unstable, making it infeasible to train the masking module for extended periods in the adversarial setting.

To address this issue, we devised a solution involving early stopping for the masking module while allowing the MAE backbone to continue learning with the increased complexity of the task. This configuration yielded improved results, surpassing the baseline in zero-shot performance by 2.9% in acc@1 and 3.6% in acc@5 metrics. However, it is important to note that this paper does not present all the results associated with this improved configuration. We are currently expanding upon the ideas proposed in this paper to explore various scenarios where we determine the ratio of adversarial training in relation to the MAE backbone.

6 Conclusion

We have introduced a novel approach for self-supervised representation learning, leveraging the concept of curriculum masking. Our method involves generating masks of increasing complexity using a novel masking module. To facilitate this progression, we have employed a progressive training strategy, wherein the masking module is trained to generate masks with increasing complexity. The progressive training translates to adversarial training after η epochs. In order to facilitate the adversarial training part, we propose a unique training strategy to support the framework of curriculum learning. We further discovered that even with a very small value for the adversarial factor, we are able to enhance the complexity of the generated masks and consequently improve the robustness of the learned representations.

References

[1] 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 *Proceedings of the ICLR*, 2021.

[2] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of ICML*, 2021, pp. 10 347–10 357.

[3] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing Convolutions to Vision Transformers," in *Proceedings of ICCV*, 2021, pp. 22–31.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of ECCV*, 2020, pp. 213–229.

[5] M. Zheng, P. Gao, X. Wang, H. Li, and H. Dong, "End-to-end object detection with adaptive clustering transformer," in *Proceedings of BMVC*, 2020.

[6] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable Transformers for End-to-End Object Detection," in *Proceedings of ICLR*, 2020.

[7] G. Ghiasi, X. Gu, Y. Cui, and T.-Y. Lin, "Scaling open-vocabulary image segmentation with image-level labels," in *Proceedings of the ECCV*, 2021.

[8] J. Qin, J. Wu, P. Yan, M. Li, R. Yuxi, X. Xiao, Y. Wang, R. Wang, S. Wen, X. Pan, *et al.*, "Freeseg: Unified, universal and open-vocabulary image segmentation," in *Proceeding of the CVPR*, 2023.

[9] X. Wang, Z. Yu, S. De Mello, J. Kautz, A. Anandkumar, C. Shen, and J. M. Alvarez, "Freesolo: Learning to segment objects without annotations," in *Proceedings of the CVPR*, 2022, pp. 14 176–14 186.

[10] A. Das, Y. Xian, Y. He, Z. Akata, and B. Schiele, "Urban scene semantic segmentation with low-cost coarse annotation," in *Proceedings of the WACV*, 2023, pp. 5978–5987.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, 2019, pp. 4171–4186.

[12] H. Bao, L. Dong, S. Piao, and F. Wei, "BEit: BERT pre-training of image transformers," in *Proceeding of ICLR*, 2022.

[13] X. Dong, J. Bao, T. Zhang, D. Chen, W. Zhang, L. Yuan, D. Chen, F. Wen, N. Yu, and B. Guo, "Peco: Perceptual codebook for bert pre-training of vision transformers," in *Proceedings of AAAI*, 2023.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of CVPR*, 2016, pp. 770–778.

[15] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, "Simmim: A simple framework for masked image modeling," in *Proceedings of the CVPR*, June 2022, pp. 9653–9663.

[16] G. Li, H. Zheng, D. Liu, C. Wang, B. Su, and C. Zheng, "Semmae: Semantic-guided masking for learning masked autoencoders," in *Proceedings of the NeurIPS*, 2022.

[17] C. Wei, H. Fan, S. Xie, C. Wu, A. L. Yuille, and C. Feichtenhofer, "Masked feature prediction for self-supervised visual pre-training," in *Proceeding of the CVPR*, 2022, pp. 14 648–14 658.

[18] Q. Zhang, Y. Wang, and Y. Wang, "How mask matters: Towards theoretical understandings of masked autoencoders," *ArXiv*, 2022.

[19] K. Chen, Z. Liu, L. Hong, H. Xu, Z. Li, and D.-Y. Yeung, "Mixed autoencoder for self-supervised visual representation learning," in *Proceeding of the CVPR*, 2023, pp. 22 742–22 751.

[20] L. Zhili, K. Chen, J. Han, H. Lanqing, H. Xu, Z. Li, and J. Kwok, "Task-customized masked autoencoder via mixture of cluster-conditional experts," in *Proceedings of the ICLR*, 2023.

[21] Y. Shi, N. Siddharth, P. Torr, and A. R. Kosiorek, "Adversarial masking for self-supervised learning," in *Proceedings of the ICML*, 2022, pp. 20 026–20 040.

[22] H. Chen, W. Zhang, Y. Wang, and X. Yang, "Improving masked autoencoders by learning where to mask," *ArXiv*, 2023.

[23] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, "Image BERT pre-training with online tokenizer," in *Proceedings of the ICLR*, 2022.

[24] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of CVPR*, 2022, pp. 16 000–16 009.

[25] R. Wang, D. Chen, Z. Wu, Y. Chen, X. Dai, M. Liu, L. Yuan, and Y.-G. Jiang, "Masked video distillation: Rethinking masked feature modeling for self-supervised video representation learning," in *Proceedings of the CVPR*, 2023, pp. 6312–6322.

[26] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, “Boosting self-supervised learning via knowledge transfer,” in *Proceedings of the CVPR*, June 2018.

[27] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proceedings of the CVPR*, June 2016.

[28] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *Proceedings of the ECCV*, 2016.

[29] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proceedings of the ICCV*, 2015.

[30] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, A. Schwarzschild, A. G. Wilson, J. Geiping, Q. Garrido, P. Fernandez, A. Bar, H. Pirsiavash, Y. LeCun, and M. Goldblum, “A cookbook of self-supervised learning,” *arXiv*, 2023.

[31] X. Li, W. Wang, L. Yang, and J. Yang, “Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality,” *ArXiv*, 2022.

[32] Y. Shi, M. Larson, and C. M. Jonker, “Recurrent neural network language model adaptation with curriculum learning,” *Comput. Speech Lang.*, vol. 33, pp. 136–154, 2015.

[33] I. Kakogeorgiou, S. Gidaris, B. Psomas, Y. Avrithis, A. Bursuc, K. Karantzalos, and N. Komodakis, “What to hide from your students: Attention-guided masked image modeling,” in *Proceedings of the ECCV*, 2022, pp. 300–318.

[34] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of ICML*, 2009, pp. 41–48.

[35] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the CVPR*, 2020, pp. 9729–9738.

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *Proceedings of the ICML*, 2020, pp. 1597–1607.

[37] 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,” in *Proceedings of the NeurIPS*, 2020, pp. 21 271–21 284.

[38] P. O. O Pinheiro, A. Almahairi, R. Benmalek, F. Golemo, and A. C. Courville, "Unsupervised learning of dense visual representations," in *Proceedings of the NeurIPS*, 2020, pp. 4489–4500.

[39] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, and L. V. Gool, "Revisiting contrastive methods for unsupervised learning of visual representations," in *Proceedings of the NeurIPS*, 2021, pp. 16 238–16 250.

[40] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *Proceedings of the ICML*, 2021, pp. 8821–8831.

[41] X. Chen and A. K. Gupta, "Webly supervised learning of convolutional networks," *Proceeding of the ICCV*, pp. 1431–1439, 2015.

[42] R. T. Ionescu, B. Alexe, M. Leordeanu, M. C. Popescu, D. P. Papadopoulos, and V. Ferrari, "How hard can it be? estimating the difficulty of visual search in an image," *Proceeding of the CVPR*, pp. 2157–2166, 2016.

[43] A. Pentina, V. Sharmanska, and C. H. Lampert, "Curriculum learning of multiple tasks," *Proceeding of the CVPR*, pp. 5492–5500, 2014.

[44] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the CVPR*, 2016, pp. 761–769.

[45] S. Braun, D. Neil, and S.-C. Liu, "A curriculum learning method for improved noise robustness in automatic speech recognition," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2017, pp. 548–552.

[46] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas, "Curriculum deepsdf," in *Proceedings of the ECCV*, 2020, pp. 51–67.

[47] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceeding of the ICML*, 2017, p. 2304–2313.

[48] D. Weinshall, G. Cohen, and D. Amir, "Curriculum learning by transfer learning: Theory and experiments with deep networks," in *Proceedings of ICML*, 2018, pp. 5238–5246.

[49] G. Hacohen and D. Weinshall, "On the power of curriculum learning in training deep networks," in *Proceedings of ICML*, 2019, pp. 2535–2544.

[50] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proceeding of the NeurIPS*, vol. 23, 2010.

[51] V. I. Spitkovsky, H. Alshawi, and D. Jurafsky, “From baby steps to leapfrog: How “less is more” in unsupervised dependency parsing,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2010, pp. 751–759.

[52] E. A. Platanios, O. Stretcu, G. Neubig, B. Póczos, and T. M. Mitchell, “Competence-based curriculum learning for neural machine translation,” in *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

ISSN (online): 2446-1628
ISBN (online): 978-87-7573-688-1

AALBORG UNIVERSITY PRESS