UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Davi Duarte de Paula

Human-Centric Self-Supervised Video Robbery Detection

Rio Claro - SP

2026

UNIVERSIDADE ESTADUAL PAULISTA

“Júlio de Mesquita Filho”

Instituto de Geociências e Ciências Exatas

Câmpus de Rio Claro

Davi Duarte de Paula

Human-Centric Self-Supervised Video Robbery Detection

Tese de Doutorado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Denis Henrique Pinheiro Salvadeo

Rio Claro - SP
2026

P324h

Paula, Davi Duarte de
Human-centric self-supervised video robbery detection / Davi Duarte de Paula. -- Rio Claro, 2026
128 p. : il., tabs.

Tese (doutorado) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientador: Denis Henrique Pinheiro Salvadeo

1. Inteligência Artificial. 2. Detecção de Roubos em Vídeos. 3. Aprendizado Profundo. 4. Visão Computacional. 5. Aprendizado Autossupervisionado. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

Impacto Potencial Desta Pesquisa

Esta pesquisa contribui para o desenvolvimento de cidades mais seguras e inteligentes ao propor técnicas de detecção automática de roubos em vídeos de vigilância baseadas em Aprendizado Profundo. Ao reduzir a necessidade de dados rotulados e aumentar a eficiência do videomonitoramento, o trabalho apoia os ODS 11 e 16 da ONU, promovendo inovação tecnológica, segurança pública e melhor uso de recursos humanos e computacionais.

Potential Impact of This Research

This research contributes to developing safer and smarter cities by proposing deep learning-based methods for automatic robbery detection in surveillance videos. By reducing the need for labeled data and improving monitoring efficiency, this work supports UN SDGs 11 and 16, promoting technological innovation, public safety, and more efficient use of human and computational resources.

UNIVERSIDADE ESTADUAL PAULISTA
“Júlio de Mesquita Filho”
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Davi Duarte de Paula

Human-Centric Self-Supervised Video Robbery Detection

Tese de Doutorado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista “Júlio de Mesquita Filho”, como parte dos requisitos para obtenção do título de Doutor em Ciência da Computação.

Comissão Examinadora

Prof. Dr. DENIS HENRIQUE PINHEIRO SALVADEO
IGCE / UNESP / Rio Claro (SP)

Prof. Dr. FABRICIO APARECIDO BREVE
IGCE / UNESP / Rio Claro (SP)

Prof. Dr. CAETANO MAZZONI RANIERI
IGCE / UNESP / Rio Claro (SP)

Prof. Dr. DAVID MENOTTI GOMES
Universidade Federal do Paraná / Curitiba (PR)

Prof Dr. WILLIAM ROBSON SCHWARTZ
Universidade Federal de Minas Gerais / Belo Horizonte (MG)

Conceito: Aprovado.

Rio Claro (SP), 22 de maio de 2025.

Acknowledgements

I thank God for strength and perseverance throughout this journey. I am deeply grateful to my family for their constant support and encouragement.

I sincerely thank my advisor, Denis Henrique Pinheiro Salvadeo, for his guidance and valuable scientific contributions. I also thank the students of the PRISMA laboratory at UNESP for their ideas, discussions, and shared moments that made this work possible.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Resumo

No estado de São Paulo, Brasil, foram registrados 228.028 roubos somente em 2023, de acordo com a Secretaria de Segurança Pública do Estado de São Paulo. Motivado por essa alta taxa de criminalidade, este trabalho tem como objetivo detectar roubos em imagens de videomonitoramento utilizando Deep Learning. A principal hipótese desta tese é que a extração das dinâmicas interpessoais das pessoas no vídeo pode melhorar o desempenho da detecção de roubos e reduzir a necessidade de dados rotulados. Para validar essa hipótese, este trabalho aplica Aprendizado Autossupervisionado (Self-Supervised Learning) nos dados de postura humana, em vez de utilizar pixels brutos dos vídeos. Até onde sabemos, não houve investigações anteriores sobre a extração de dinâmicas interpessoais dentro da área de pesquisa definida por estes três tópicos: (i) Aprendizado Autossupervisionado, (ii) Postura Humana e (iii) Detecção de Roubos. O objetivo final deste trabalho foi desenvolver soluções de videomonitoramento que melhorem a qualidade da detecção de roubos, exigindo menos dados rotulados para treinamento, representando um avanço em relação à análise manual tradicional de vídeos, que é demorada e ineficiente. Para a realização da pesquisa, foi criado um novo conjunto de dados chamado CamNuvem dataset, contendo vídeos reais de situações de roubo, com anotações tanto em nível de quadro quanto em nível de vídeo. Um benchmark inicial foi realizado, comparando diversos métodos de detecção de anomalias em vídeo. A primeira contribuição deste trabalho é uma tarefa pretexto de Aprendizado Autossupervisionado que recebe os pixels dos vídeos como entrada e tenta prever as futuras posições e posturas dos indivíduos na cena. O objetivo dessa tarefa pretexto é aprender as relações entre as pessoas. O treinamento e os testes foram realizados utilizando os conjuntos de dados de detecção de anomalias CamNuvem e UCF-Crime, com o uso de um extrator de características I3D tradicional. Em 12 cenários experimentais, 7 deles (58%) apresentaram melhores resultados do que o baseline (detecção de anomalias sem a tarefa pretexto proposta), sendo que o melhor caso alcançou 88,40% de AUC no conjunto de dados CamNuvem, uma melhora de 2,6% em relação à linha de base. A segunda contribuição é outra tarefa pretexto de Aprendizado Autossupervisionado que utiliza apenas as posturas das pessoas em uma cena para prever suas posturas futuras, sem depender dos pixels brutos dos vídeos. Neste caso, no entanto, é necessário um passo de pré-processamento para detectar e identificar os indivíduos no vídeo. Entre os diversos cenários experimentais avaliados, o melhor resultado demonstrou uma melhora de 4,01% em AUC em relação ao baseline utilizando o conjunto de dados CamNuvem, ao mesmo tempo que exigiu significativamente menos dados e nenhuma

anotação feita por humanos. Embora esses resultados mostrem benefícios ao usar a abordagem proposta, experimentos mostraram que não é possível afirmar que essa melhora é resultado direto da extração de relações interpessoais nos vídeos. Portanto, não foi possível confirmar a nossa hipótese: o uso de Aprendizado Autossupervisionado para extrair dinâmicas interpessoais a partir da postura humana é útil para melhorar a qualidade da detecção de roubos. Mesmo assim, as abordagens apresentadas neste trabalho podem ser aplicadas em qualquer tarefa contendo vídeos exibindo interações entre pessoas, e, no futuro, elas podem ser avaliadas em aplicações como análise de estratégias em esportes coletivos.

Palavras-chave: Aprendizado Autossupervisionado, Videomonitoramento, Detecção de Roubos, Aprendizado Profundo, Detecção de Anomalias em Vídeos

Abstract

In São Paulo State, Brazil, 228,028 robberies were reported in 2023 alone, according to the São Paulo State Public Security Department. Motivated by this high crime rate, this work aims to detect robberies in video surveillance footage using Deep Learning. The main hypothesis of this thesis is that extracting the interpersonal dynamics of people in the video can improve robbery detection performance and reduce the need for labeled data. To validate this hypothesis, this work applies Self-Supervised Learning to human posture data, rather than using raw video pixels. To the best of our knowledge, there has been no prior investigation into extracting interpersonal dynamics within the research area defined by these three topics: (i) Self-Supervised Learning, (ii) Human Posture, and (iii) Robbery Detection. The ultimate goal of this work is to develop surveillance solutions that enhance robbery detection quality while requiring less labeled data for training, offering an improvement over traditional manual video analysis, which is both time-consuming and inefficient. For the research, a novel dataset named CamNuvem was created, containing real-world videos of robbery situations, with both frame-level and video-level annotations. An initial benchmark was conducted comparing several video anomaly detection methods. The first contribution of this work is a Self-Supervised Learning pretext task that takes raw video frames as input and attempts to predict the future positions and postures of individuals in the scene. The aim of this pretext task is to learn the relationships among people. Training and testing were conducted using the anomaly detection datasets CamNuvem and UCF-Crime, employing a traditional I3D feature extractor. Across 12 experimental scenarios, 7 (58%) achieved better results than the baseline (anomaly detection without the proposed pretext task), with the best case reaching 88.40% AUC on the CamNuvem dataset, an improvement of 2.6% over the baseline. The second contribution is another Self-Supervised Learning pretext task that uses only the postures of people in a scene to predict their future postures, without relying on raw video pixels. In this work, however, a preprocessing step is required to detect and identify individuals in the video. Among multiple experimental scenarios evaluated, the best result showed an improvement of 4.01% in AUC over the baseline using the CamNuvem dataset, while requiring significantly less data and no human-generated labels. Although these results show benefits from using the proposed approach, experiments indicated that it is not possible to state that this improvement is a direct result of extracting interpersonal relationships from the videos. Therefore, our hypothesis could not be confirmed: the use of Self-Supervised Learning to extract interpersonal dynamics from human posture is useful for improving the quality of theft detection. Nevertheless, the

approaches presented in this work can be applied to any task involving videos that display interactions between people, and in the future, they may be evaluated in applications such as the analysis of strategies in team sports.

Keywords: Self-Supervised Learning, Video Surveillance, Robbery Detection, Deep Learning, Video Anomaly Detection

List of abbreviations and acronyms

VAD	*Video Anomaly Detection*
EBM	*Energy-Based Model*
LVM	*Large Vision Model*
LLM	*Large Language Model*
SSL	*Self-Supervised Learning*
ANN	*Artificial Neural Network*
DNN	*Deep Neural Network*
CNN	*Convolutional Neural Network*
SSL	*Self-Supervised Learning*
GAN	*Generative Adversarial Network*
FCN	*Fully Connected Network*
UCF	*University of Central Florida*
UCSD	*University of California San Diego*
CUHK	*Chinese University of Hong Kong*
MIL	*Multiple Instance Learning*
RTFM	*Robust Temporal Feature Magnitude*
TSM	*Temporal Shift Module*

Contents

1 INTRODUCTION 14
1.1 Motivation 15
1.2 Problem 16
1.3 Contribution 18
1.4 Work Organization 19

2 RELATED WORKS 20
2.1 Self-Supervised Learning 20
2.1.1 Fundamental Concepts 22
2.1.1.1 Latent Variables 22
2.1.1.2 Training an EBM and the Concept of Collapse 23
2.1.1.3 Avoid collapse with Contrastive methods 25
2.1.1.4 Avoid collapse with Regularized methods 26
2.1.2 Architectures 27
2.1.2.1 Generative Methods 28
2.1.2.2 Joint Embedding Methods 29
2.1.2.3 Joint Embedding Predictive Architecture 35
2.1.2.4 Context-Based Methods 38
2.1.2.4.1 Context Similarity 38
2.1.2.4.2 Spatial Context Structure 38
2.1.2.4.3 Temporal Context Structure 39
2.1.2.5 Free Semantic Label-Based Methods 40
2.1.2.6 Cross Modal-Based Methods 40
2.1.2.6.1 RGB-Flow Correspondence 41
2.1.2.6.2 Visual-Audio Correspondence 41
2.1.2.6.3 Ego-Motion 41

3 DEEP LEARNING-BASED VIDEO SURVEILLANCE ANOMALY DETECTION 42
3.1 Datasets 42

3.2 Methods **45**
3.2.1 Self-Supervised Learning 45
3.2.2 Semi-Supervised Video Anomaly Detection 46
3.2.2.1 One-Class Learning 46
3.2.2.2 Interpretable Learning 47
3.2.3 Multiple Instance Learning (MIL) Approach 48
3.2.4 Weakly Supervised Video Anomaly Detection 48
3.2.4.1 Robust Temporal Feature Magnitude Learning (RTFM) (TIAN et al., 2021) . . . 49
3.2.4.2 Weakly Supervised Anomaly Localisation (WSAL) (LV et al., 2021) 50
3.2.4.3 Real-world Anomaly Detection in Surveillance (RADS) (SULTANI; CHEN; SHAH, 2018) 52
3.2.5 Fully Video Anomaly Detection 53
3.2.6 Unsupervised Video Anomaly Detection 54

4 CAMNUVEM: A ROBBERY DATASET FOR VIDEO ANOMALY DETECTION 56
4.1 CamNuvem Dataset Statistics **56**
4.2 Initial Benchmark **58**
4.2.1 Results 61
4.3 Conclusions **65**

5 SELF-SUPERVISED FEATURE EXTRACTION FOR VIDEO SURVEILLANCE ANOMALY DETECTION 66
5.1 Methodology **67**
5.1.1 Pretext Task 67
5.1.2 Downstream Task 69
5.1.3 Experimental Setup 70
5.2 Results **71**
5.2.1 General Evaluation 71
5.2.1.1 UCF-Crime dataset 72
5.2.1.2 CamNuvem dataset 73
5.3 Discussion and Conclusions **74**

6 SELF-SUPERVISED LEARNING FOR EXTRACTING INTERPERSONAL DYNAMICS IN VIDEO ROBBERY DETECTION 75

6.1 Methodology . . . 75
6.1.1 Dataset . . . 76
6.1.1.1 CamNuvem dataset . . . 76
6.1.1.2 UCF-Crime dataset . . . 78
6.1.2 Definitions . . . 79
6.1.2.1 SMPL . . . 80
6.1.2.2 Joints 3D . . . 81
6.1.2.3 SMPL + Joints 3D . . . 81
6.1.3 Pretext Task . . . 81
6.1.4 Sample Multiplication . . . 85
6.1.5 Downstream Task . . . 86
6.1.5.1 Training On Splits . . . 87
6.1.5.2 Training On The Same Set . . . 87
6.2 Results and Discussions . . . 87
6.2.1 Results for Training on Splits . . . 87
6.2.2 Results for Training On The Same Set . . . 89
6.2.3 People Relationship Study . . . 95
6.2.4 New data generation analysis . . . 97
6.3 Chapter Conclusion . . . 97

7 VALIDATING THE EFFECTIVENESS OF INTERPERSONAL DYNAMICS EXTRACTION IN THE PRETEXT TASK . . . 98

8 CONCLUSIONS . . . 106

APPENDIX A – HYPERPARAMETERS . . . 122
A.1 Raw data . . . 122
A.2 Person Quantity Analisys . . . 123

1 Introduction

A growing demand for effective surveillance and security solutions in Brazil has emerged due to the high number of crimes, particularly in urban areas. According to a recent survey conducted by the Brazilian Institute of Geography and Statistics (IBGE), 1.8 million Brazilians were victims of robbery in 2021, which equates to approximately seven robberies every two minutes (IBGE, Coordenação de Pesquisas por Amostra de Domicílios, 2022b). The survey also assessed the public's sense of insecurity, revealing that 51.7% of Brazilians are afraid to walk alone at night (IBGE, Coordenação de Pesquisas por Amostra de Domicílios, 2022a). In the state of São Paulo alone, the São Paulo State Public Security Department reported 218,839 thefts in 2020, 225,706 in 2021, 242,991 in 2022, and 228,028 in 2023.

Given this situation, it is crucial to develop advanced technological solutions to support the prevention and early identification of criminal activity. In this context, the use of surveillance cameras in both public and private areas to monitor and record the movement of people and vehicles has become increasingly common. However, manual image analysis is time-consuming and costly, often falling short in detecting anomalies and suspicious activities.

This work aims to develop an approach to analyze surveillance videos and enable the automatic detection of robberies. It is important to clarify that the term robbery in this work follows the definition in Brazilian legislation: the theft of movable property belonging to another person, for oneself or someone else, using serious threats or violence against a person (BRASIL, 1940). Therefore, the goal is to detect instances of violence or threats made by an aggressor intending to steal an item. Situations involving theft without violence (such as shoplifting) are not prioritized.

The development of such a robbery detection system may also result in several scientific contributions that are applicable to broader video understanding tasks, including object detection, semantic segmentation, predictive sports analysis (STIVAL, 2022), and video caption generation (TAN; CHAN, 2019). In the course of this chapter, Section 1.1 presents the motivation for this work, while Sections 1.2 and 1.3 detail the problem and the contributions of this thesis, respectively.

1.1 Motivation

Deep learning is the primary approach widely used in state-of-the-art methods across various language benchmarks (BROWN et al., 2020; DEVLIN et al., 2019) and computer vision benchmarks (KAY et al., 2017; DENG et al., 2009; GU et al., 2018). Recently, with the increase in computational power and the emergence of hardware specifically designed for deep learning, there has been a rise in both the size and quantity of data used by these models. This has led to the development of Large Language Models (LLMs) for language-based applications and Large Vision Models (LVMs) for computer vision tasks. These models are designed to overcome limitations that have persisted in the field of artificial intelligence for decades. However, they require expensive hardware, high training and inference costs, and vast amounts of data.

A fundamental strategy for training an LLM or an LVM is Self-Supervised Learning (SSL) (JING; TIAN, 2020), an approach that uses unlabeled data to pre-train deep learning models in a task-agnostic manner. Once pre-trained, these models can be fine-tuned for specific tasks with relatively little labeled data. Since SSL does not require human-annotated data, it enables the use of massive datasets. For example, GPT-2 (Generative Pre-trained Transformer 2) was pre-trained on the WebText dataset, which includes content from 45 million websites (RADFORD et al., 2019). Similarly, the LVM DINOv2 (Self-**Di**stillation With **No** Labels v2) (OQUAB et al., 2023) was pre-trained on the LVD-142M dataset (OQUAB et al., 2023), which consists of a mixture of image datasets and web-scraped images. Thus, SSL enables the extraction of knowledge with minimal human effort.

In the domain of human action recognition (HAR), using human posture as input to deep learning models instead of raw image pixels can significantly improve results (RAJASEGARAN et al., 2023), as posture captures the actual movements made by individuals over time. There are several ways to represent human posture, such as using 3D joint coordinates (GOEL et al., 2023), 2D joint coordinates (XU et al., 2022), or 3D mesh representations (XU et al., 2022). Leveraging the advantages of SSL, a natural next step would be to apply self-supervised learning to human posture data. This could lead to innovations in the design of pretext tasks and training methodologies, as demonstrated in works like Kocabas, Karagoz and Akbas (2019) and Ludwig et al. (2021).

When applied to the task of robbery detection, the quality of the detection could depend heavily on how people interact with each other in a scene. For example, consider a surveillance

video where a robber quickly approaches another person, and the victim subtly steps away and raises their hands. In this situation, understanding the interpersonal dynamics between the robber and the victim could enhance the semantic representation of the scene, ultimately improving the detection accuracy. While some studies have explored human-object interactions (e.g., Rajasegaran et al. (2023)), they typically rely on supervised learning to detect actions involving individuals in a scene.

In contrast, limited work has been done to explore interpersonal dynamics using self-supervised learning. Given that SSL can leverage data transformations and does not require labeled data for pretext tasks, we argue that it is possible to develop SSL-based methods that operate on human posture to extract meaningful interpersonal interactions. Therefore, the main objective of this work is to extract human interpersonal dynamics (or human relationships) from videos to improve robbery detection. Although this thesis focuses specifically on robbery detection in video surveillance, the techniques proposed here are designed to be general enough to be applicable to any video involving human interactions.

1.2 Problem

To develop a robbery detection system, a deep learning approach was adopted due to its growing popularity and its ability to extract intricate and abstract features from input data. Deep learning has proven effective in analyzing videos for pattern recognition purposes. Additionally, the self-supervised learning (SSL) approach (JING; TIAN, 2020) enables unsupervised model training without the need for manual data labeling, thereby reducing costs and reliance on large amounts of labeled data.

The visual characteristics of robbery situations are highly variable, making them difficult to define precisely. The aggressor may use different types of weapons and varying levels of violence, and surveillance cameras may be positioned at different angles. The environment can also differ significantly, ranging from streets to indoor settings. In some cases, the similarity to normal activities makes detection even more challenging, as the aggressor may attempt to conceal their intentions. Furthermore, a robbery may span a long temporal window and involve multiple individuals and complex interactions.

Supervised deep learning can be used to model such situations. However, this approach requires a large dataset, which is difficult to obtain due to the rarity of such events in video

data. Even if a sufficiently large dataset were available, the labeling process would be costly, as annotations would need to be made at the frame level to precisely mark the beginning and end of each robbery event.

To address these challenges, a classical approach is to consider anomaly detection as a reconstruction problem (SINGH; PANKAJAKSHAN, 2018; DOSHI; YILMAZ, 2021; WANG et al., 2019). These methods learn the distribution of normal videos by reconstructing them. Since only normal videos are used to train the reconstructor, the model generates a high reconstruction error when presented with a video containing a robbery, indicating an anomaly. Over time, various methods have targeted different aspects of videos to improve anomaly detection. Some classical approaches rely on human tracking to detect anomalies (JIANG et al., 2011), but unlike the approach proposed in this work, they do not consider interpersonal dynamics, only individual human behavior. This often leads to poor performance due to occlusions in scenes (SABOKROU et al., 2018a). Other methods extract motion features using optical flow, which captures movement by subtracting one frame from another (WANG et al., 2019; RAMCHANDRAN; SANGAIAH, 2020; DOSHI; YILMAZ, 2020).

However, these methods often perform poorly in real-world scenarios. To improve real-world anomaly detection, Sultani, Chen and Shah (2018) introduced the Multiple Instance Learning (MIL) framework, which includes abnormal samples during training, thereby improving the effectiveness of surveillance-based anomaly detection. Building on this foundation, multiple studies have explored MIL-based alternatives (TIAN et al., 2021; LV et al., 2021). In MIL, anomaly samples are weakly labeled: only a binary label indicates whether a video contains an anomaly, while the exact start and end points remain unknown. This significantly reduces the human annotation effort required.

Another approach to mitigate the scarcity of labeled data and reduce the need for manual annotations is self-supervised learning (SSL). SSL enables pre-training a model on a large unlabeled dataset to capture fundamental data structures. The model can then be fine-tuned using a smaller labeled dataset, achieving strong performance even with limited supervision. Despite its advantages, training state-of-the-art SSL models is computationally expensive. To address this, our approach explores using human posture as input instead of raw pixels. This reduces the data dimensionality, accelerates training, and allows the use of modest computational resources. While this requires a preprocessing step to extract human poses, it needs to be performed only once before training. Moreover, the current state-of-the-art action recognition method on

Table 1 – Publications resulting from this thesis.

Number	Title	Type	Local	Status
1	CamNuvem: A robbery dataset for video anomaly detection	Journal	MDPI Sensors	Published
2	Self-Supervised Feature Extraction for Video Surveillance Anomaly Detection	Conference	SIBGRAPI	Published
3	Self-Supervised Learning for Extracting Interpersonal Dynamics in Video Robbery Detection	Journal	Springer Nature Sn Computer Science	Published

the AVA (Atomic Visual Actions) dataset (GU et al., 2018) is based on human posture data (RAJASEGARAN et al., 2023), further justifying this approach.

Most video surveillance anomaly detection methods use benchmarks with a variety of anomaly types, such as car accidents, general violence, and assaults. Although the widely used UCF (University of Central Florida) Crime dataset (SULTANI; CHEN; SHAH, 2018) includes a robbery category, it contains few examples and lacks a focused effort on robbery detection in video surveillance contexts.

Given this context, the central hypothesis of this work is: Is it possible to improve video surveillance robbery detection by extracting interpersonal relationships? This will be pursued using SSL, a proven technique for learning visual representations without labeled data. Human posture will be used instead of raw pixel input, as it directly encodes human movement and enables training under limited computational resources. This hypothesis is innovative, as no prior work has used SSL on human posture data to explicitly extract interpersonal dynamics for improving robbery detection in surveillance videos.

1.3 Contribution

In order to validate our hypothesis presented in the previous Section, two papers have been published, and one is currently under evaluation. These works are listed in Table 1.

In the first publication, the challenges of data acquisition and preparation were addressed. This work proposed the creation of a large, fully annotated video surveillance robbery dataset, which facilitated research and the development of methods for robbery detection. The second publication introduced a self-supervised learning approach that takes raw video frames as input to

a deep learning model but incorporates person trajectories during training to extract interpersonal dynamics. The results were obtained from two different datasets and evaluated within the context of video anomaly detection. The third presented a different approach that takes trajectories as input and uses SSL on this type of data. The results were evaluated in both supervised robbery detection and supervised anomaly detection tasks. The findings from this study demonstrate that the central hypothesis of this thesis could not be confirmed. It was not possible to affirm that extracting the interpersonal dynamics of people in the videos can improve robbery detection performance and reduce the need for labeled data. However, the data and methodologies proposed in the published papers remain valid and useful for future work.

1.4 Work Organization

To guide the reader through this work, Chapter 2 will review related works on self-supervised learning. Chapter 3 will discuss about surveillance anomaly detection. Chapter 4 will introduce the new dataset proposed, which includes fully labeled videos for robbery detection and is used to train and validate the methods discussed. Chapters 5 and 6 will present the first and second self-supervised methods for robbery detection proposed in this work, respectively. Chapter 7 describes how we evaluated the method proposed in this thesis and concluded that it does not take into account the interpersonal dynamics of people in the videos, thus not allowing us to confirm our hypothesis. Finally, Chapter 8 will discuss the results obtained and conclude the work.

2 Related Works

As stated before, this work aims to explore how to use Self-Supervised Learning to enable a DNN to learn to detect robberies in video surveillance footage by utilizing pose and other feature vectors pre-extracted from the humans in the video. Thus, Section 2.1 presents a review of the main approaches in the Self-Supervised Learning research field and Section 3 discusses existing Video Anomaly Detection methods.

This chapter presents an introduction to Self-Supervised Learning (Section 2.1) and Deep Learning-based Video Surveillance Anomaly Detection (3).

2.1 Self-Supervised Learning

The machine learning tasks that use DNNs typically require a large labeled dataset, which is generally hard to obtain. However, sometimes there is a large dataset with only a few labels available. Thus, Self-Supervised Learning (SSL) offers a way to extend the labeled samples **automatically** using information inherent to the dataset. These labels are created in a task known as a **pretext task**, aiming to **learn representations of the dataset**. The learned representations can then be transferred to another task, known as a **downstream task**, which typically employs conventional Supervised Learning on the labeled portion of the dataset, such as action classification, semantic classification, or object detection. It is important to note that semi-supervised learning aims to use the available labels in the labeled portion to estimate the labels in the unlabeled portion, while self-supervised learning focuses on extending the labeled dataset (even if weakly annotated) through controlled transformations of the data, automatically generating labels for this extended portion. Subsequently, this newly labeled data is used to train a DNN that extracts the characteristics of the dataset, which can be applied to other machine learning tasks.

Self-Supervised Learning (SSL) is a learning paradigm in the field of machine learning that aims to use non-human-labeled data in the training pipeline of a model, unlike fully-supervised methods that rely explicitly on human-labeled data. This approach alleviates the costs of human annotation and thereby enables the use of a large quantity of data. The SSL training pipeline is divided into two stages: pretext task training and downstream task training.

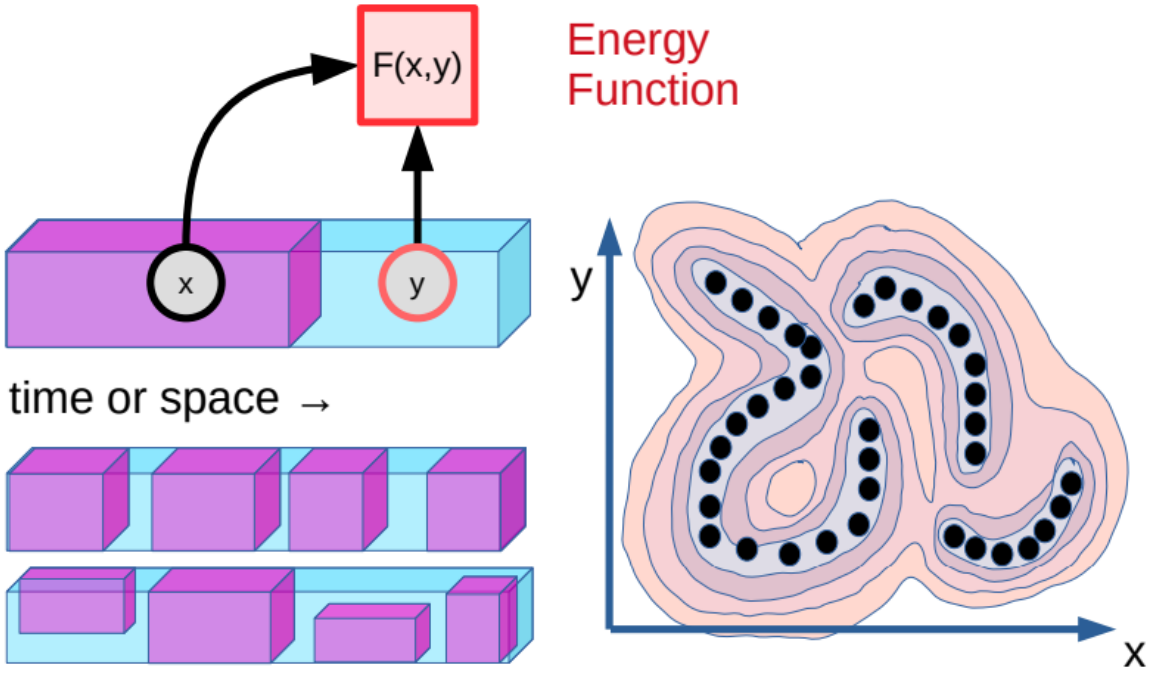
Suppose there is a large dataset of unlabeled surveillance videos and a small portion of the data (e.g., 1% of the data) is labeled by humans. Suppose also the main objective (downstream task) is to detect video segments where a robbery occurs. The pretext task involves using the unlabeled data to extract the essential structure of the dataset; its objective is not to solve the pretext task. Labels are created artificially based on the structure of the videos, and the model is trained in a fully supervised manner using these artificially generated labels. These labels can vary, and numerous studies explore different methods for generating them. More generally, the labels are created by occluding part of the dataset and predicting this occluded part using the remaining visible sections. This approach allows the model to learn the intrinsic structure of the dataset and the inductive bias[1] (MITCHELL, 1980).

The formulation of the pretext task can be described using an Energy-Based Model (EBM), which is defined as a function $F(x, y)$ that assigns low energy (value) when x and y are compatible and high energy when x and y are not compatible. The variable x can be a visible part of a video, such as past frames, and the variable y can be occluded parts of a video, such as future frames. Figure 1 shows an example of such a scheme. On the left, a sample from a dataset is represented, where purple indicates the visible part and blue represents the occluded part. No human label is needed in this case. The sample could be of any type, involving either time or space dimensions, and the visible and occluded parts could be located anywhere. On the right, the energy landscape is represented. The EBM $F(x, y)$ assigns low energy for compatible x and y, represented by the gray color, while moving away from these points results in higher energy, represented by the orange color. In the video dataset scenario, each sample has a time axis, where the visible part x would be past frames and the occluded part y would be future frames. The EBM $F(x, y)$ would assign a low value if the future frames y are from the same video as x, and a high value if y is from a different video than x (incompatible).

Using the example of video robbery detection, if $F(x, y)$ were trained in this way, it would be able to predict representations of future video given representations from the past. This could allow the function F to acquire a set of hierarchical abstract representations of how the world works (LECUN, 2022). This knowledge could improve the performance of the model for the downstream task. The general training procedure is as follows: given two variables x and y, learn two functions, $s_x = g_x(x)$ and $s_y = g_y(y)$, such that s_x and s_y are maximally informative about x and y, respectively, and s_y can be easily predicted from s_x. In this way, note that the EBM

[1] Inductive bias refers to the assumptions a model makes in order to output a value for unseen data.

Figure 1 – Left: EBM scheme. Right: Visualization of the energy levels.



Source: LeCun (2022).

operates in the latent space: no raw pixels are predicted.

After, it time to fine-tune the model pre-trained in the pretext task in the downstream taks, e.g. video robbery detection. Now the model will be fine-tuned using the scarcy samples labelled by human, that in the video robbery detection example represents only 1% of the dataset. The ideia is that, using the knowledge acquired on the pre-training made in the pretext task, the model could achieve a better performance in the downstream task (robbery detection).

2.1.1 Fundamental Concepts

Section 2.1.1.1 introduces the concept of latent variables, Section 2.1.1.2 explains the phenomenon of collapse, Section 2.1.1.3 discusses how to avoid collapse using contrastive methods, and Section 2.1.1.4 describes how to prevent collapse with regularized methods.

2.1.1.1 Latent Variables

There are some cases that involve uncertainty, where some information in y cannot be predicted using only x, and cases where multiple different y values could be predicted using the same x. In these cases, the uncertainty could be represented as a latent variable, and the

prediction can use this value to encode the missing information or to predict a specific output y given by this variable. The latent variable can be seen as a parameter to choose one element in the set of possible relationships between x and y.

Imagine a person entering a shop, where there are two equally probable paths they could take: turning left or turning right. Suppose a surveillance camera is filming this person. If x is a frame of the video before the person chooses a path, and y is a frame of the video after the choice, the compatibility of x and y depends on a latent variable that encodes whether the person turned left or right. In such temporal cases, the latent variable represents all the information that cannot be encoded by x (the past) and is needed to predict y (the future).

In this way, an Energy-Based Model (EBM) E_w, based on learned parameters w, that depends on x, y, and a latent variable z is called a Latent-Variable EBM (LVEBM)

$$E_w(x, y, z). \tag{2.1}$$

Using an optimization process, one can find a value of the latent variable z that minimizes the energy. This is known as the inference procedure for the EBM:

$$\hat{z} = \arg\min_{z \in \mathcal{Z}} E_w(x, y, z), \tag{2.2}$$

Since the value of z that minimizes the energy in the EBM is known, one can remove this variable from the equation:

$$F_w(x, y) = \min_{z \in \mathcal{Z}} E_w(x, y, z) = E_w(x, y, \hat{z}), \tag{2.3}$$

where $F_w(x, y)$ is called zero-temperature free energy.

2.1.1.2 Training an EBM and the Concept of Collapse

The training procedure must seek a vector w that parameterizes a model $F_w(x, y)$ (e.g., a deep learning model). In this way, the training procedure will optimize the energy surface of the model. The training procedure has to make $F_w(x, y)$ assign low energy to compatible x and y pairs, and high energy to incompatible x and $\hat{y}$ pairs.

Considering a training dataset containing compatible samples (x, y) and incompatible samples $(x, \hat{y})$, one way to train the model is by using a suitable loss function $\mathcal{L}$ that makes

the energy of $F_w(x, y)$ lower than the energy of all other values $\hat{y}$ existing in the dataset, e.g., $F_w(x, \hat{y})$. However, the main dificult is to ensure that $F_w(x, \hat{y}) > F_w(x, y)$, for all $\hat{y} \neq y$. Without any provision, the EBM training can suffer from **collapse**, when the energy landscape can become "flat", meaning the model $F_w(x, y)$ might assign the same energy to all values of y.

The occurrence of collapse during training depends on the architecture. Figure 2 shows a few examples of architectures. Figure 2 a) depicts a deterministic generative architecture, in which x and y are samples (e.g., images), with y being predictable from x. Here, $s_x = \text{Enc}(x)$ is an encoder (e.g., a deep learning model) that aims to extract the important knowledge from x and ignore the irrelevant details for prediction, making s_x a latent representation of x. Then, another model $\tilde{y} = \text{Pred}(s_x)$ (e.g., a deep learning model) tries to reconstruct y from s_x. If x and y are images, this reconstruction is performed in the raw pixel domain, justifying the architecture's name "generative". $D(y, \tilde{y})$ is the loss function used to train $\text{Pred}(s_x)$ and $\text{Enc}(x)$. In this type of architecture, collapse is not possible as long as $x \neq y$ and $D(y, \tilde{y})$ is strictly greater than zero in these cases. The only case where the energy can be zero is when $y = \tilde{y}$.

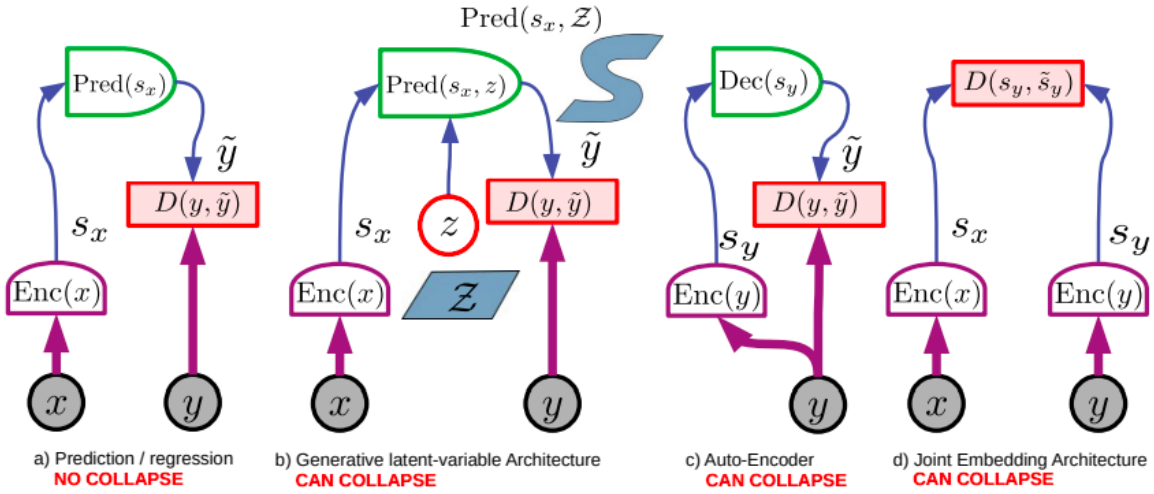
In the generative latent-variable architecture shown in Figure 2 b), collapse can occur if the latent variable z has too much informational capability. The prediction $\tilde{y}$ varies as z varies over the set $\mathcal{Z}$, in which $\tilde{y}$ would match the y compatible with x. However, if z is too "large", the region of low energy can be larger than the region of high data density. For example, if z has the same dimension as y, the prediction $\text{Pred}(s_x, z)$ could assign zero energy to the entire y space.

In the auto-encoder architecture shown in Figure 2 c), collapse can occur if the latent representation of y, s_y, has too much informational capability. If s_y has the same or higher dimension size than y, then the decoder, represented by $\text{Dec}(s_y)$, can effectively output the identity function, producing a reconstruction error equal to zero for all y inputs.

In the joint embedding architecture shown in Figure 2 d), collapse can occur if the encoders $s_x = \text{Enc}(x)$ and $s_y = \text{Enc}(y)$ ignore the inputs x and y, producing constant values for the outputs such that $s_x = s_y$. This happens when the informational capability of s_x and s_y is insufficient, e.g., vectors with too low a dimension size.

To avoid the collapse in any architecture, one can choose between two approaches, contrastive and regularized methods. Contrastive have a problem for scaling up the training data while regularized method is preferred (LECUN, 2022). These two methods will be discussed on Sections 2.1.1.3 and 2.1.1.4, respectively.

Figure 2 – Different architectures and their possibility to collapse in training.



Source: LeCun (2022).

2.1.1.3 Avoid collapse with Contrastive methods

Training contrastive methods means using a loss function that, in the minimization process, makes the model F_w assign a low energy to a sample (x, y) and a high energy to a sample $(x, \hat{y})$, where $\hat{y}$ is suitably selected to not be compatible with x, in contrast to y. Selecting a $\hat{y}$ that is not compatible with x is equivalent to selecting a sample from outside the region of low energy. In this way, the minimization process can estimate the low and high energy surfaces using the y and $\hat{y}$ pairs in the training process. Figure 3, in the top-right part, presents a visual representation of the contrastive setup. The green dots represent the contrastive samples, e.g., $\hat{y}$, and the blue dots represent the compatible samples, e.g., y. The training process estimates the low energy surface (orange area) using the boundaries between compatible and contrastive samples.

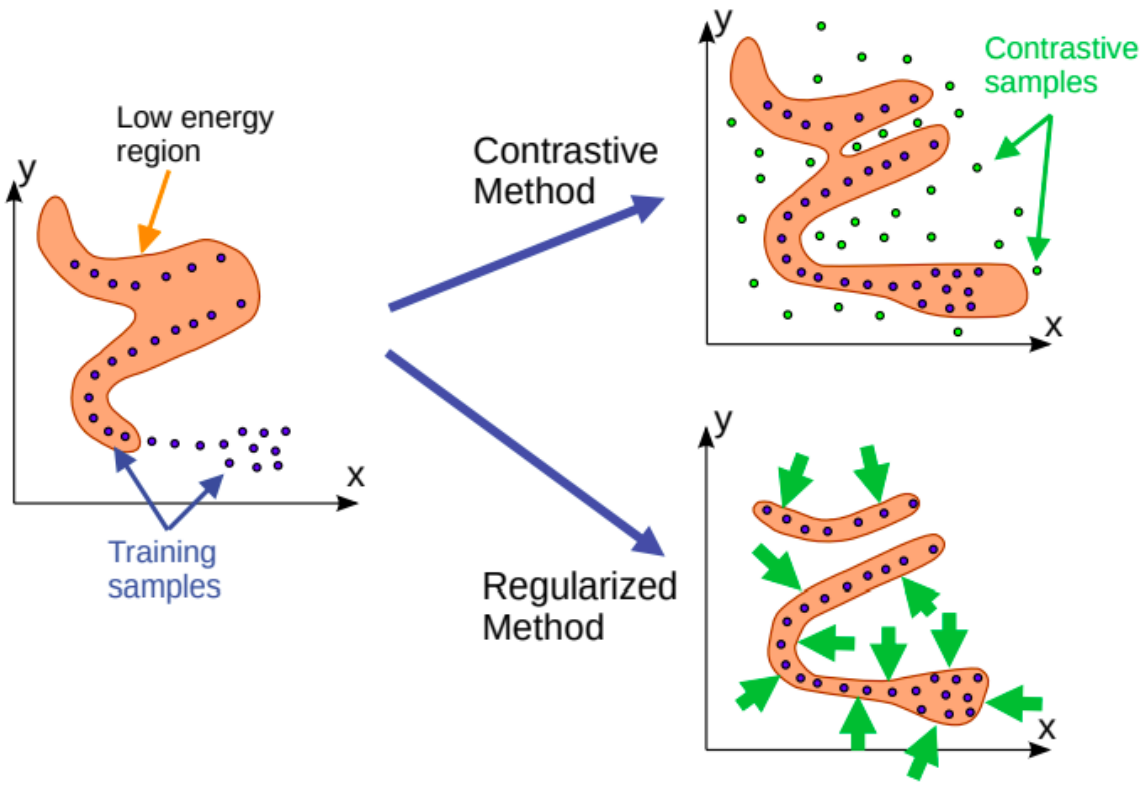
The loss function used in the contrastive minimization has to be an increasing function of $F_w(x, y)$ and a decreasing function of $F_w(x, \hat{y})$. For example, in the following equation, the loss function H is contrastive (LECUN, 2022):

$$\mathcal{L}(w, x, y, \hat{y}) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y})), \tag{2.4}$$

where $\mathcal{L}$ increases when $F_w(x, y)$ increases and decreases when $F_w(x, \hat{y})$ decreases, as long as the latter is less than the former plus $m(y, \hat{y})$ (a positive margin). The Hinge Loss (ROSASCO et al., 2004; BOSER; GUYON; VAPNIK, 1992) is an example of such a loss:

$$\mathcal{L}(w, x, y, \hat{y}) = [F_w(x, y) - F_w(x, \hat{y}) + \mu||y - \hat{y}||]^+, \tag{2.5}$$

Figure 3 – Visualization of contrastive and regularized methods.



Source: LeCun (2022).

where $[b]^+$ is identity when b is positive and zero otherwise.

These types of loss functions are often used in Siamese networks (CHICCO, 2021; DU; FANG; SHEN, 2017), where y is a corrupted or distorted version of x, and $\hat{y}$ is a random sample selected to be an incompatible sample with x. A problem with this approach is that there needs to be an appropriate strategy to select $\hat{y}$ to contrast with x. Another issue is that it may be necessary to have a large number of contrastive examples $\hat{y}$ to cover all the high-energy regions of the hyperspace where the data distribution of x cannot reach. Because of this, the number of contrastive examples can grow exponentially as the dimensionality of the representation increases, which is a reason to avoid these types of methods (LECUN, 2022).

2.1.1.4 Avoid collapse with Regularized methods

To avoid the problem of dimensionality, regularized methods to training EBMs use a loss function that reduces the energy of training samples while simultaneously minimizing the shape of the multidimensional region associated with the low-energy y samples. This is achieved by minimizing a regularization term in the energy/loss function, which makes the methods less

susceptible to the dimensionality problem. The problem, however, is designing such regularizers, and the solution depends on the model architecture. Nevertheless, note that contrastive and regularized methods are not disjoint; they are independent concepts that may be used together.

Using the methods described in Figure 2, one can identify appropriate ways to design regularized solutions. For the Generative latent-variable architecture (Figure 2 (b)), one solution is to limit the latent variable z, thereby consequently limiting the shape of the low-energy latent space. For example, if z is a discrete latent variable with k values, then the latent space y will have at most k points with zero energy. If z is a vector with dimension size d, then the shape of the space with zero energy in the latent space y will have at most a dimension size of d.

In the Auto-encoder architecture, shown in Figure 2 (c), regularization can be achieved by limiting the dimension size of s_y, which will restrict its informational capacity. Similar to the Generative latent-variable architecture, this will restrict the shape of the region in y with low energy, thereby avoiding collapse.

In the Joint Embedding Architecture, shown in Figure 2 (d), one way to avoid collapse is to use a loss function that explicitly maximizes the information in s_x about x and maximizes the information in s_y about y, thereby decreasing the volume of the region with low energy in y.

2.1.2 Architectures

The design of the pretext task represents the main step in Self-Supervised Learning. Some works present different perspectives on the organization of the major types of SSL pretext tasks. LeCun (2022) divides the methods based on the loss function, classifying them into (i) contrastive and (ii) regularized. On the other hand, other works, such as Garrido et al. (2024) and Jing and Tian (2020), classify the approaches based on the model architecture, which is the classification used in this section.

Section 2.1.2.1 describes the Generative methods, Section 2.1.2.2 describes the Joint Embedding methods, Section 2.1.2.3 describes the Joint Embedding Predictive methods, Section 2.1.2.4 describes the Context-Based methods, Section 2.1.2.5 will describe the Free Semantic Label-Based Methods, and Section 2.1.2.6 will describe the Cross Modal-Based methods.

2.1.2.1 Generative Methods

Generative methods are those that generate an entire image in the pixel space at some point in the pretext task pipeline. The model architecture or the task used for that is work-dependent, and some of these methods are explained below:

- **Image Generation with GAN:** It uses Generative Adversarial Networks (GOODFELLOW et al., 2014) (GAN), which consist of two DNNs. The first one, named Generator, tries to generate a fake image from latent vectors. The second one, named the Discriminator, attempts to detect whether an image presented as input is a fake image (generated by the Generator) or a real image (from the dataset). Therefore, both fake and real images are presented as input for the Discriminator. With training, both DNNs will improve in their tasks, and the Discriminator will learn the dataset distribution. After this step is concluded, transfer learning from the Discriminator to a downstream task is performed. Note that it is not necessary to have human-generated labels, as only the dataset is needed to distinguish between fake and real images;

- **Super-Resolution:** Instead of using a GAN to generate an entire image, the work by Ledig et al. (2017) aims to generate a high-resolution image from a low-resolution image. Therefore, the Generator receives a low-resolution image as input and tries to upsample it. This is based on the assumption that a set of high-resolution images is available and that the low-resolution images are generated from them. Thus, there is no need for human-generated labels;

- **Image Inpainting:** Instead of using a GAN to generate a high-resolution image from a low-resolution one, this category aims to inpaint an image from a degraded one (PATHAK et al., 2016). This is based on the assumption that there is a dataset of perfect images, and the degraded images are generated from them, usually by hiding a small patch in their content. Therefore, the degraded images are input to the Generator, and the Discriminator needs to detect whether an image is the original or the reconstructed one. Note that no human-generated labels are needed. Another approach is to use a fully convolutional network to generate the high-resolution image, with the degraded image as input and the original image as the label (PATHAK et al., 2016);

- **Image Colorization:** A set of color images can be converted in grayscale to generate the

inputs, meaning that the labels (original images) are already available without human effort. A DNN architecture used is a fully convolutional network with L2 Loss; the input is the decolorized images, and the label is the original images (ZHANG; ISOLA; EFROS, 2016);

- **Video Generation:** The same principles applied in Image Generation with GAN can be extended to videos. In the work of Vondrick, Pirsiavash and Torralba (2016), they utilized two networks as part of the Generator: one to generate moving objects and another to generate the background and static objects. Both DNNs generate video segments from a random noise vector. Thus, the Discriminator aims to identify the generated video segments and the real video segments. Note that no human-generated labels are used. Ultimately, the weights of the Generator can be transferred to a downstream task;

- **Video Colorization:** The same principles applied in Image Colorization can be extended to videos. Given a set of video frames, they can be transformed into grayscale, and a DNN can be employed to predict their RGB pixel values, since these values are already known in the original colorized dataset. Thus, an input-output sample is generated without human effort. Several pretext task approaches can be developed. In the work of Vondrick et al. (2018), they utilized video colorization as a supervisory signal to achieve visual tracking. On the other hand, the work of Tran et al. (2016) used an encoder-decoder DNN based on U-Net (RONNEBERGER; FISCHER; BROX, 2015), where the input is a set of frames in grayscale and the output is the corresponding set of frames in RGB;

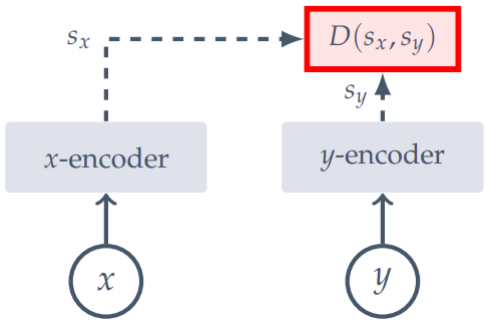
- **Video Future Prediction:** Given any dataset, a pretext task can be designed by enclosing a part or segment of the samples in this dataset and using a DNN to predict this hidden part. In a video dataset, a DNN can predict a future frame given a set of past ones or predict a past frame given a set of future ones (SRIVASTAVA; MANSIMOV; SALAKHUDINOV, 2015; VILLEGAS et al., 2017; MATHIEU; COUPRIE; LECUN, 2015). In the work of (VILLEGAS et al., 2017), they used an LSTM to encode the temporal features of frames and an encoder-decoder architecture to encode the spatial features of frames.

2.1.2.2 Joint Embedding Methods

The Joint Embedding methods aim to augment a sample using some technique (e.g., random crop on images, color distortion, etc.) and predict the latent representation of an augmented version given the latent representation of another augmented view. Unlike Generative

methods, Joint Embedding methods do not have to generate the full image in pixel space, which would require significant computational effort. Furthermore, the model doesn't need to expend effort generating small details in the final image, which may not be necessary for representation learning. Figure 4 shows the general Joint Embedding architecture. The x and y are two different augmented versions of the same image, and they are respectively passed through encoders x-encoder and y-encoder, creating s_x and s_y. The objective in the training process is for s_x to be as equal as possible to s_y, using the loss function $D(s_x, s_y)$. Some of the Joint Embedding methods are described below:

Figure 4 – Joint-Embedding Architecture.

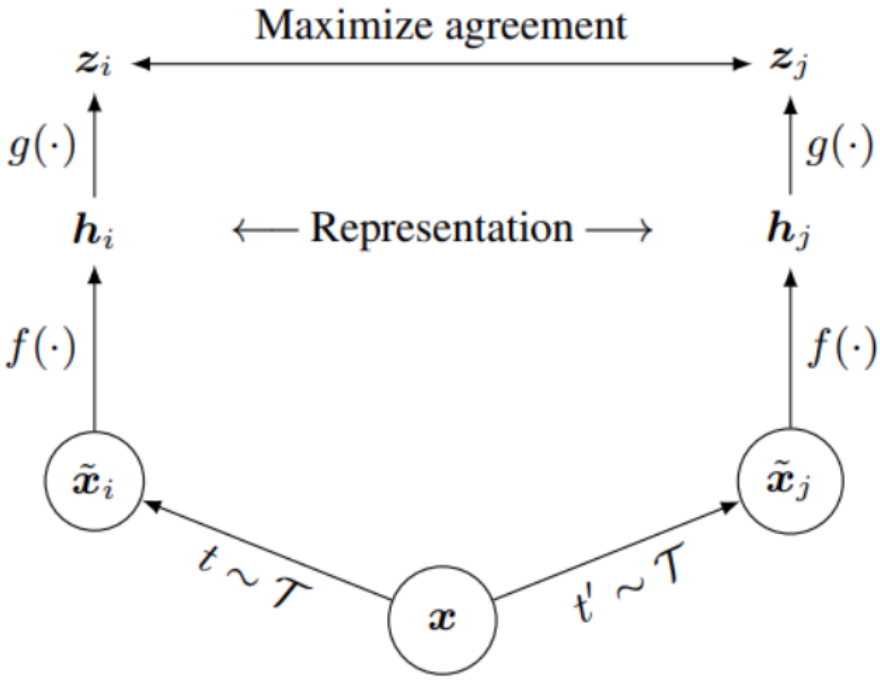


Source: Assran et al. (2023)

.

- **SimCLR (CHEN et al., 2020a)**:

Figure 5 – SimCLR architecture.



Source: Chen et al. (2020a).

Given an image x, two augmented views of the same image are generated, $\tilde{x}_i$ and $\tilde{x}_j$, produced by, for example, random crop, resizing, random color distortion, etc. A neural network $f(.)$ is applied over $\tilde{x}_i$ and $\tilde{x}_j$, obtaining a representation h_i and h_j, respectivelly. After, a projector $g(.)$ is applied over h_i and h_j, resulting in the latent representation z_i and z_j, respectivelly. Using a constrative loss, the z_i and z_j that are inferred from the same image have their distance minimized, while the representation inferred from different images have their distance augmented. In the downstream task, the projector $g(.)$ is throw away and only the encoder $f(.)$ is used. Figure 5 shows the described process.

This work also contributed with a study of the image augmented techniques for $\tilde{x}_i$ and $\tilde{x}_j$, and find that choosing the right technique plays a critical role in the performance of the method. They found that using just one augmentation not performed very well, it is necessary to use a set of augmentations. The set that presented the best result is the color distortion plus random crop.

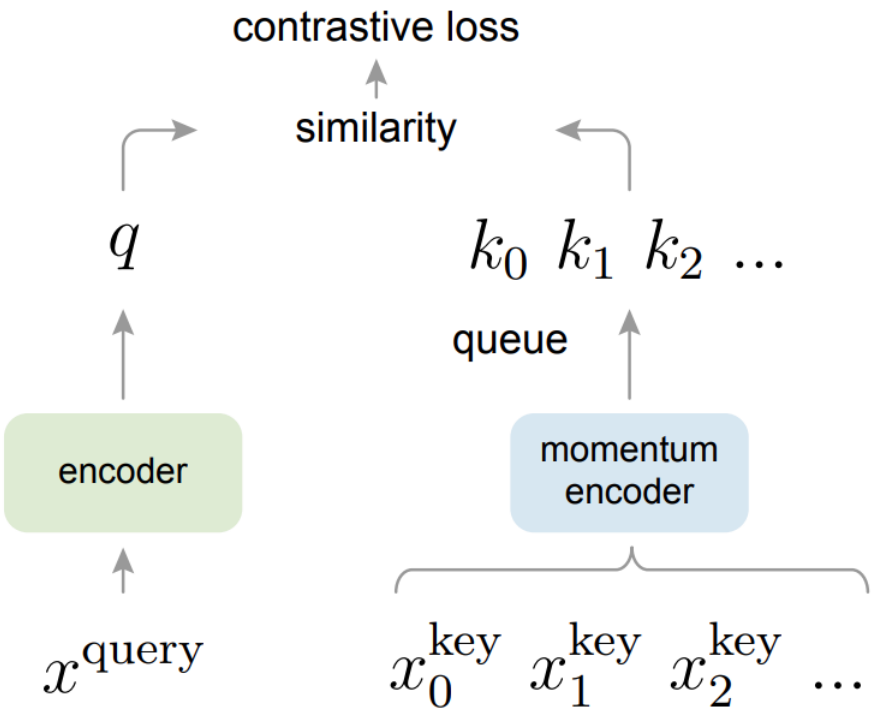
Furthermore, this work identified that the layer $f(.)$ represents better the information than the projector $g(.)$, because the loss function in the pretext task makes $g(.)$ specialized in the pretext task, that not represent the downstream task. In this way, the layer before $g(.)$,

the $f(.)$, acquires more general representation, what is more transferable, and so more suitable to use in downstream tasks.

Finally, the work observed in their results that big batch sizes and more training steps benefits the results, in comparison with supervised deep learning. This is compatible with the fact that a contrastive method need several samples to be compared to construct representation manifold;

- **MOCO (HE et al., 2020)**:

Figure 6 – MOCO architecture.



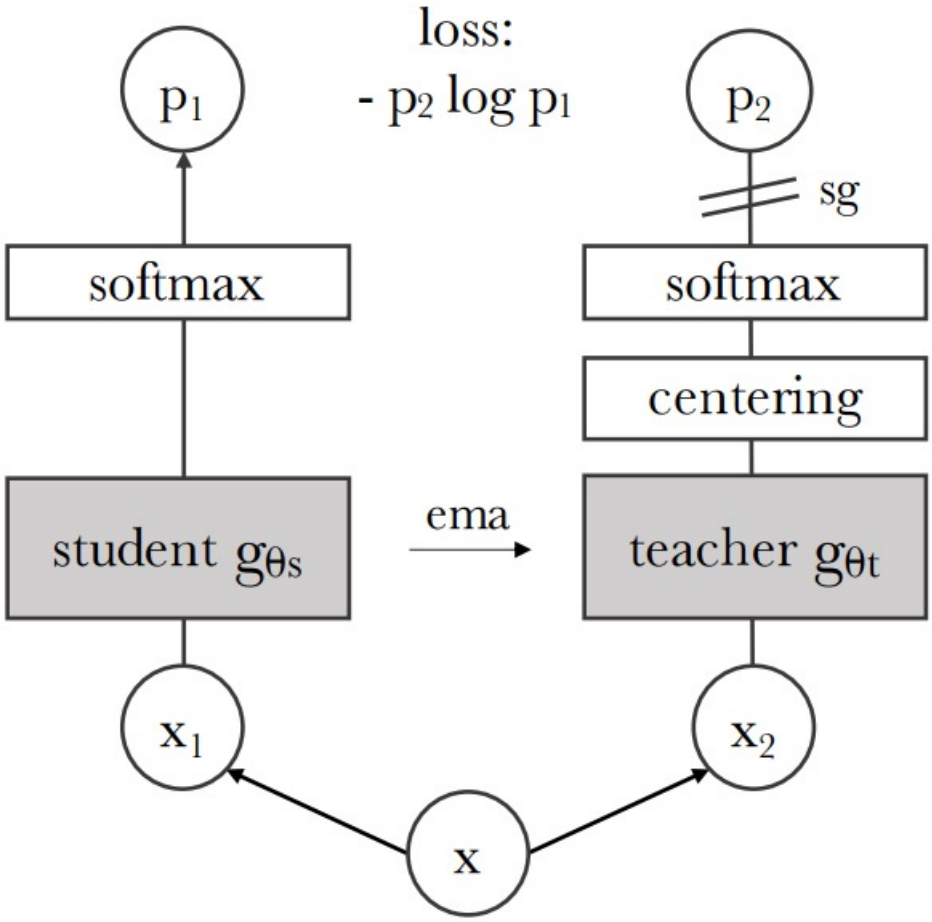
Source: He et al. (2020).

The SimCLR method requires large batch sizes and more training steps compared to supervised deep learning. In contrast, the Momentum Contrast (MOCO) (HE et al., 2020) method aims to shift the negative examples from the batch to a queue that stores negative examples. In the training step, the incompatible examples $\hat{y}$ used to contrast with x are stored in a queue filled with examples from the immediate preceding mini-batches. However, updating the key encoder model via backpropagation using all negative examples is intractable, because the gradient must propagate to all samples. To deal with this problem, the work proposes making the key encoder non-trainable, updating its weights using a

momentum update over the weights from the online model. Figure 6 shows the architecture of this model;

- **DINO (CARON et al., 2021):**

Figure 7 – DINO architecture.



Source: Caron et al. (2021).

The work of Caron et al. (2021) builds the model named self-**di**stillation with **no** labels (DINO), and the general architecture is shown in Figure 7, where x is an image and x_1 and x_2 are two augmented views of the same image. Note that the model $teacher_{g\theta_t}$ is not trainable (as denoted by the symbol **sg**, meaning stop gradient), but its weights are updated using an Exponential Moving Average (EMA), similar to MOCO (HE et al., 2020).

Among their main contributions, they explore different architectural components, in which their impact on downstream scores in the validation set of the ImageNet dataset is shown in Table 2. The architectural componentes evaluated are Momentum Update (Mom.) (HE et

al., 2020), Sinkhorn-Knopp (SK) (DISTANCES, 2013; CARON et al., 2020), Multi-Crop (MC) augmentation on the augmented views x_1 and x_2, different loss functions (CE is Cross Entropy, MSE is Mean Square Error, and INCE is InfoNCE) and a predictor layer after the student network as made in Grill et al. (2020) (Pred.). In Table 2, the seventh column (**k**-NN) shows the accuracy of the model's frozen features (after pretext task training) using the **k**-NN classifier (without training) in the ImageNet validation. The second to sixth columns show the architectural components evaluated. Note that in lines 7, 8, and 9, the methods BYOL (Bootstrap Your Own Latent) (GRILL et al., 2020), MoCov2 (Momentum Contrast v2) (CHEN et al., 2020b), and SwAV (Swapping Assignments between Views) (CARON et al., 2020) are also compared.

Table 2 – Comparison of different methods with key attributes.

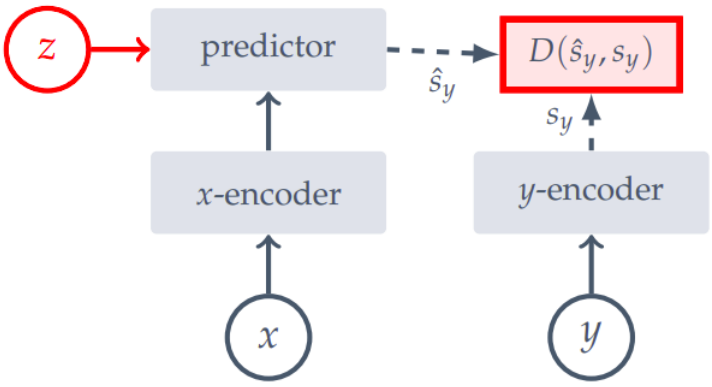
Method	**Mom.**	**SK**	**MC**	**Loss**	**Pred.**	**k-NN**
DINO	✓	✗	✓	CE	✗	72.8
”	✗	✗	✓	CE	✗	0.1
”	✓	✓	✓	CE	✗	72.2
”	✓	✗	✗	CE	✗	67.9
”	✓	✗	✓	MSE	✗	52.6
”	✓	✗	✗	CE	✓	71.8
BYOL	✓	✗	✗	MSE	✓	66.6
MoCov2	✓	✗	✗	INCE	✗	62.0
SwAV	✗	✓	✓	CE	✗	64.7

Source: Caron et al. (2021).

The reader can note, after analysing Table 2, that DINO, using Momentum Update, Multi-Crop, and Cross-Entropy as loss functions, achieved a significant improvement compared to the other methods.

2.1.2.3 Joint Embedding Predictive Architecture

Figure 8 – Joint-Embedding Predictive Architecture.

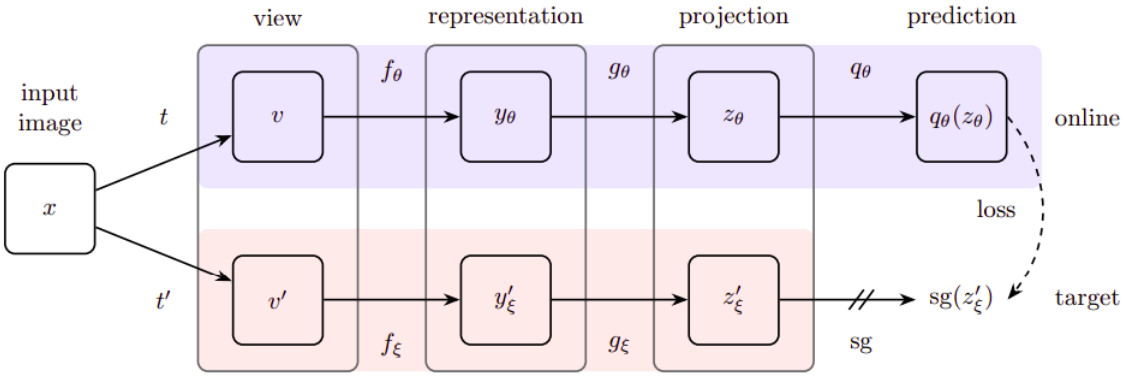


Source: Assran et al. (2023).

The Joint-Embedding Predictive Architecture (JEPA), as shown in Figure 8, is composed of an encoder $s_y = y\text{-encoder}(y)$ that receives an augmented view y of an image and outputs a latent representation s_y, and an encoder followed by a predictor $\hat{s}_y = \text{predictor}(x\text{-encoder}(x), z)$, which takes as input another augmented view x of an image and outputs the estimate $\hat{s}_y$ of s_y. Note the presence of the variable z as an input to the predictor. The presence of this predictor represents the main difference from the Joint-Embedding Architecture (Figure 4).

- **BYOL (GRILL et al., 2020)**:

Figure 9 – BYOL architecture.



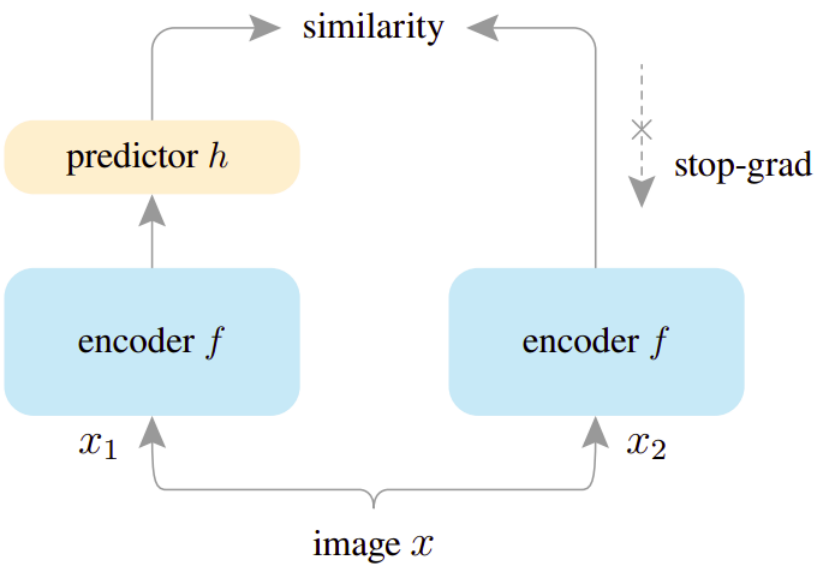
Source: Grill et al. (2020).

The Bootstrap Your Own Latent (BYOL) (GRILL et al., 2020) architecture, shown in Figure 9, is not contrastive; it uses a Mean Square Error loss between the output of the

online network and the output of the target network. In this image, the augmented views of the image x are depicted as v and v', where the first is input to the online network and the second to the target network. The target network is not trainable, as indicated by the sg (stop gradient) annotation in Figure 9, but its weights are updated using an Exponential Moving Average from the online network, as done in other works (HE et al., 2020; CARON et al., 2021). To regularize the training and avoid the problem known as collapse, this work employs an asymmetry between the online and target networks by using a prediction layer at the end of the online network, which does not exist in the target network. In the downstream task, the only part used is the encoder, represented by the layer y_θ in Figure 9;

- **SimSiam (CHEN; HE, 2021)**:

Figure 10 – SimSiam architecture.



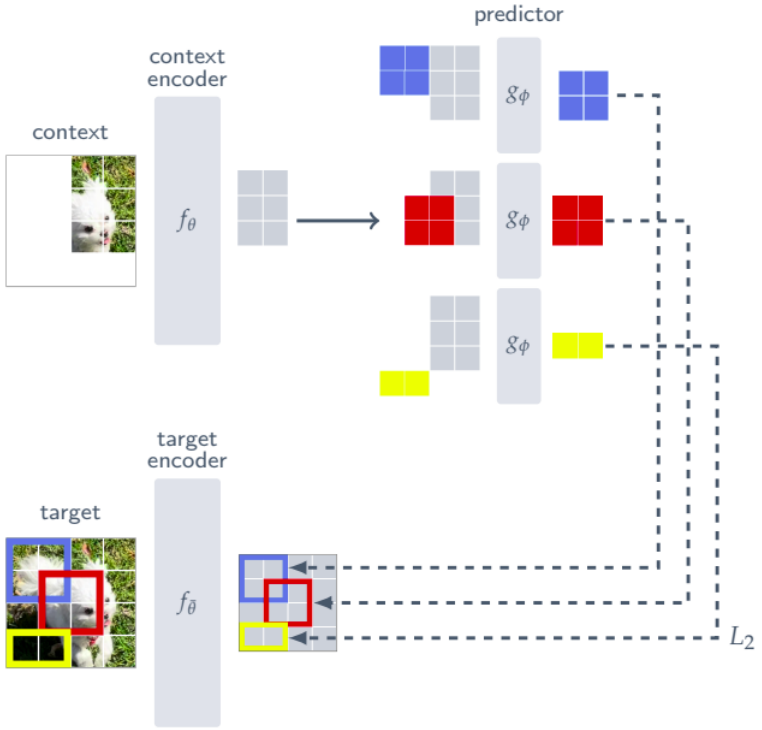
Source: Chen and He (2021).

The **Sim**ple **Siam**ese networks (SimSiam) can be seen as a "BYOL without the momentum encoder" (CHEN; HE, 2021). Its architecture is shown in Figure 10. The augmented views x_1 and x_2 of an image x are input to the model encoder f. The output of the first encoder f is passed through another layer named predictor h; the work found this asymmetry essential for converging to good results. Although in the BYOL work the momentum encoder was defined as essential to avoid collapse, SimSiam achieved good results without it, using only stop gradient, as can be seen in Figure 10. The weights of the two encoders shown in this figure are the same. It does not use contrastive loss, but an L_2 loss between the output of the predictor h and the encoder f. In addition, this work showed that it does not depend on

large batch sizes, which reduces the dependency on large GPU (Graphics Processing Unit) memory during the training stage, unlike the contrastive works of SimCLR (CHEN et al., 2020a) and SwAV (CARON et al., 2020), which require large batch sizes (e.g., 4096);

- **I-JEPA (ASSRAN et al., 2023)**:

Figure 11 – I-JEPA architecture.



Source: Assran et al. (2023).

The Image-based Joint-Embedding Predictive Architecture (I-JEPA) is designed to predict the latent representations of patches from an augmented view of an image x, based on the latent representations of patches from another augmented view of the same image. Figure 11 illustrates this architecture, where f_θ serves as the encoder, mapping the augmented views of x into latent space. This approach avoids operating in pixel space, thereby eliminating irrelevant details and focusing on meaningful representations for learning.

The target encoder mirrors the architecture of the context encoder, and its weights are updated using an Exponential Moving Average of the weights from the context encoder. In the context encoder, part of the latent representation is occluded using special mask tokens, which are fed into the predictor g_ϕ as latent variables. The latent representation are input to the predictor in the usual manner.

The output of the predictor is compared with the latent representation generated by the target encoder using a suitable loss function. This loss is then utilized to update both the projector g_ϕ and the encoder f_θ.

In downstream tasks, the projector is discarded, leaving only the context encoder for inference. Notably, I-JEPA employs a non-contrastive loss function, it utilizes a straightforward L_2 distance between the predicted and target representations, which simplifies the training process and reduces computational overhead.

2.1.2.4 Context-Based Methods

Considering that some parts of a dataset are moved from their correct places, the pretext tasks in the Context-Based category aim to predict their correct locations. Note the difference from the Generation-Based pretext tasks (Section 2.1.2.1), which aim to generate parts of the dataset that are hidden.

2.1.2.4.1 Context Similarity

- **Clustering:** Clustering is an unsupervised task applied when several data points need to be grouped together according to their similarity. In the image context, a feature vector represents an image in a multi-dimensional space. Images with feature vectors in the same cluster represent the most similar ones, while images with feature vectors in different clusters represent the most dissimilar images.

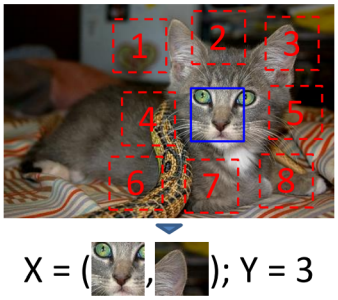
  In the Self-Supervised Learning domain, a pretext task could be designed to use a CNN to extract feature vectors from images. The feature vectors are then clustered (without supervision), and the cluster assignments are used as supervision signals. Thus, the CNN used to extract the feature vectors has its weights updated during the training phase based on the cluster supervision. Furthermore, the weights of the CNN are transferred to the downstream task. Notable works in this category include Mundhenk, Ho and Chen (2018), Li et al. (2016), Caron et al. (2018).

2.1.2.4.2 Spatial Context Structure

- **Image Jigsaw Puzzle:** The pretext task in this category aims to predict a specific part of an image using another part, as described in Doersch, Gupta and Efros (2015). Figure 12a

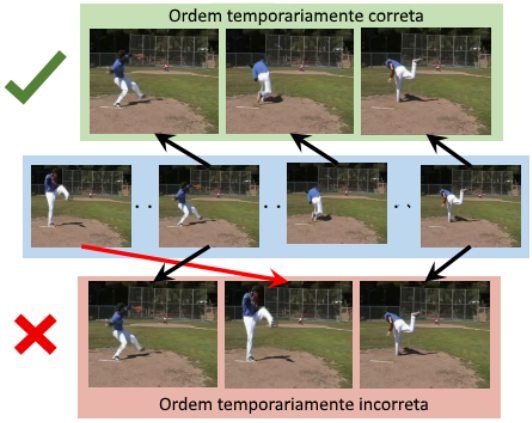
Figure 12 – Different approaches for automatic label generation in the pretext task.

(a) SSL approach that uses image segment prediction in the pretext task.



Source: Doersch, Gupta and Efros (2015).

(b) SSL approach that uses temporal verification of video frames in the pretext task.



Source: Misra, Zitnick and Hebert (2016).

shows an example of this approach. Given the input of the cat's eye segment, a DNN must predict the correct position of the ear, which is in the upper right corner, corresponding to the third segment;

- **Geometric Transformation:** Another approach for the pretext task consists of rotating a video at K discrete angles and using the rotation value as a supervision signal (JING et al., 2018). This improves spatio-temporal understanding and the performance of DNNs on video recognition datasets with small sample sizes. This approach can also be applied to images (GIDARIS; SINGH; KOMODAKIS, 2018).

2.1.2.4.3 Temporal Context Structure

- **Frame Order Verification:** When the dataset consists of videos, another pretext task is to verify whether a set of frames is in the correct order, as demonstrated in the work of Misra, Zitnick and Hebert (2016), illustrated in Figure 12b. In this Figure, the center line represents a set of frames extracted from a video that is not necessarily in order, while the top line represents the correct frame order, and the bottom line represents an incorrect order. In this case, the idea of the pretext task is to reorder the video frames. Note that the labels are already available due to the dataset structure, as the original videos intrinsically contain the correctly ordered frames. Thus, no human effort is needed to create the labels;

- **Frame Order Recognition:** Another way to construct a pretext task is through Space-Time Cubic Puzzles (KIM; CHO; KWEON, 2019), where the spatio-temporal structure of the video is divided into cubes, and a 3D CNN is employed to place the cubes in their respective positions. This is a direct extension of the work by Doersch, Gupta and Efros (2015) into the spatio-temporal domain.

2.1.2.5 Free Semantic Label-Based Methods

When the labels of a dataset are synthetically generated using game engines like AirSim (SHAH et al., 2018) and Carla (DOSOVITSKIY et al., 2017), it is still considered Self-Supervised Learning, as no human labor is employed to generate the labels. These tools can generate images with abundant information about depth, contours, surface normals, segmentation masks, and optical flow (JING; TIAN, 2020) for artificial scenes or environments, which can be used as labels. Therefore, a DNN trained on fully synthetic images may not perform well in a realistic environment, necessitating additional techniques to bridge the gap between synthetic and realistic data. For example, the work of Ren and Lee (2018) employed a DNN to predict surface normals, depth, and instance contours for synthetic images, while real images were utilized in an adversarial architecture to address the disparity between synthetic and real images.

Another way to automatically generate labels is by applying hard-coded programs that create salience and foreground masks (such as foreground object segmentation (PATHAK et al., 2017)), contours/edge detection (LI et al., 2016), and depth maps for images and videos (JIANG et al., 2018). This approach can yield a high volume of labels. It is similar to the Knowledge Distillation technique because, regardless of the hard-coded method used to generate labels, the main objective is to distill the knowledge embedded in them. Knowledge Distillation methods aim to directly use the output of an already trained DNN (or the output of intermediate layers) as labels, while in this context, the labels are generated by methods not based on Deep Learning.

2.1.2.6 Cross Modal-Based Methods

This category of pretext tasks aims to find relationships between two different datasets or sensors. Note the difference between the pretext tasks in the Context-Based category (Section 2.1.2.4), which aim to find the correct position of some part of the dataset.

2.1.2.6.1 RGB-Flow Correspondence

Considering a video, the frame flow estimation encodes the information of moving objects and correlations about them. The RGB information encodes the appearance of static frames. An alternative pretext task is to verify whether the RGB information corresponds with the flow information. To achieve this, the work of (SAYED; BRATTOLI; OMMER, 2018) employed two individual DNNs—one to extract features from RGB input and another to extract features from the flow estimation. After that, another DNN finds correspondence between these two DNNs (one applied in the RGB domain and another in the flow feature domain). Therefore, this third DNN encodes valuable invariant knowledge between these two domains.

2.1.2.6.2 Visual-Audio Correspondence

When the dataset contains multi-modal data, such as video and audio, a pretext task can be employed to find references between them. The work of Korbar, Tran and Torresani (2018) uses a DNN to find correspondences between single frames or sets of stacked frames and the Fast Fourier Transform of the audio. The authors created positive data, which are samples of audio and video extracted from the same temporal point, and negative data, which are samples of audio and video that do not correspond to the same temporal point. Therefore, the DNN has to learn the correspondence between the audio and video to discover the positive samples.

2.1.2.6.3 Ego-Motion

The previous section discussed how different modalities of data, acquired with different sensors, can be utilized to design a pretext task (such as image and audio). The knowledge learned by a DNN is the invariant knowledge between these two sensors, which would be very valuable as a starting point for a downstream task. In this way, one can design a pretext task using *ego-motion*. This term refers to the 3D movement of the camera within the scene. Thus, this type of pretext task uses the correlation between the apparent image movement and the actual movement made by the camera (acquired via an additional odometry sensor). There are several domains that use cameras that move around an environment, such as self-driving cars and robotic applications. Therefore, the work of Agrawal, Carreira and Malik (2015) utilizes a set of images as input to a DNN, with the real translation and rotation movements made by the camera as output.

3 Deep Learning-based Video Surveillance Anomaly Detection

Video Surveillance Anomaly Detection is a research area that learns the normal patterns of video surveillance footage and detects anomalies (or abnormalities), which are moments when the video deviates from the normal pattern. These anomalies do not have a clear definition; they can vary according to the application domain, such as car crashes, violence, or people falling. However, this approach can be employed to detect robberies, which is the main objective of this work. Therefore, this chapter collects the main advancements in Video Surveillance Anomaly Detection methods. Section 3.1 discusses the main Video Surveillance datasets for anomaly detection. Section 3.2 discusses the main methods and approaches to detect anomalies in surveillance videos.

3.1 Datasets

Aiming to improve real-world anomaly detection, Sultani, Chen and Shah (2018) proposed a dataset called UCF-Crime, which is composed of 1,900 real-world video clips totaling 128 hours, and a method that uses multiple instance learning to find samples with abnormalities for the training process. This method decreases the cost of labeling, since the dataset only has to be weakly labeled, i.e., each video only needs to be annotated in two classes (video-level annotation) to specify whether or not an anomaly exists in the video. There is no need for frame-level annotation.

Other datasets have been proposed in the literature to enable the development of video surveillance anomaly detection methods, but these datasets are limited in terms of camera variation, resolution, point of view, weather conditions, etc. They offer alternative benchmarks for current academic works on video surveillance anomaly detection. The UCSD dataset (LI; MAHADEVAN; VASCONCELOS, 2013) contains videos with a few types of anomalies, such as people walking in unusual paths and locations. The Avenue dataset (LU; SHI; JIA, 2013) presents videos from the CUHK campus avenue. The Boss dataset (VELASTIN; GÓMEZ-LIRA, 2017) includes videos and audio from multiple cameras capturing people's behavior inside a

moving train. The Subway Entrance and Subway Exit datasets (ADAM et al., 2008) contain videos from the entrance and exit of a subway station, respectively.

The CamNuvem dataset (PAULA; SALVADEO; ARAUJO, 2022), proposed in this work and explained in more detail in the publication depicted in Chapter 4, along with its subsequent version, the CamNuvem Balanced dataset, explained in more details in the publication depicted in Chapter 6, contain videos of robberies captured from real-world cameras. Table 3 presents a comparison of several datasets in terms of the video quantity, average frame number, and types of anomalies in the datasets.

Table 3 – Comparison of several video surveillance anomaly detection datasets. * Values in parentheses refer to the novel robbery samples only.

	Number of Videos	**Average Number of Frames**	**Dataset Length**	**Anomaly Examples**
Avenue (LU; SHI; JIA, 2013)	37	839	30 min	Bikers, small carts, walking across walkways
BOSS (VELASTIN; GÓMEZ-LIRA, 2017)	12	4052	27 min	Bikers, small carts, walking across walkways
Subway Entrance (ADAM et al., 2008)	1	121,749	1.5 h	Wrong direction, no payment
Subway Exit (ADAM et al., 2008)	1	64,901	1.5 h	Wrong direction, no payment
UCSD Ped1 (LI; MAHADEVAN; VASCONCELOS, 2013)	70	201	5 min	Run, throw, new object
UCSD Ped2 (LI; MAHADEVAN; VASCONCELOS, 2013)	28	163	5 min	Run
UMN (RAGHAVENDRA; BUE; CRISTANI, 2006)	5	1290	5 min	Harass, disease, panic
UCF–Crime (SULTANI; CHEN; SHAH, 2018)	1900	7247	128 h	Abuse, arrest, arson, assault, accident, burglary, fighting, robbery
UCF–Crime Extension (ÖZTÜRK; CAN, 2021)	240	3060	7.5 h	Protest, violence using Molotov cocktails, fighting
CamNuvem (PAULA; SALVADEO; ARAUJO, 2022)	972 (486) *	6329 (2303)*	57 (10.4)* hours	Robbery
CamNuvem Balanced	972 (486)*	1314 (2303)*	11.8 (10.4)* hours	Robbery

Source Sultani, Chen and Shah (2018).

3.2 Methods

In this Section the main approaches for deep learning-based video surveillance anomaly detection is discussed.

This Section is divided as follows: Section 3.2.1 will discuss the methods based on the Self-Supervised Learning approach. Section 3.2.2 will discuss the methods based on the Semi-Supervised approach. Section 3.2.4 will discuss the methods based on the Weakly Supervised approach. In this work, the terms Video Surveillance Anomaly Detection and Video Anomaly Detection (VAD) will be used interchangeably here.

3.2.1 Self-Supervised Learning

The main problem of SSL is how to design the pretext task, and this section will discuss some designs applied to Video Anomaly Detection. In this section, for simplicity, the various types of VAD methods are divided into Reconstruction and Prediction, but they could be further divided into the SSL categories already explained in Section 2.1.

- **Reconstruction:** This approach designs its pretext task to reconstruct some part of the video sequence given another part. The part to be predicted can be occluded or degraded, e.g., using noise corruption (SUN et al., 2021; CHEN et al., 2021; CONG; YUAN; LIU, 2011), patch inpainting (RISTEA et al., 2024; MADAN et al., 2024; RISTEA et al., 2022), frame occlusion, etc. The objective of the model is to predict the correct version of the degraded part. The pretext task can also reconstruct part of the video using only encode-decode operations (HASAN et al., 2016; LUO; LIU; GAO, 2017; HUANG et al., 2021), without explicitly degrading part of the videos.

- **Prediction:** Unlike the former category of pretext tasks, the prediction does not aim to reconstruct frames or patches in the pixel domain but to predict any element of the video without relying on raw pixels. Given historical data from a video, the objective is to predict the data at the current moment. This type of pretext task can alleviate the high computational cost of reconstructing frames or patches in the pixel domain, in addition to allowing the model to focus on larger features rather than small details in the raw frame reconstruction, which may not improve the final anomaly detection accuracy. Some examples of such works include the prediction of person trajectories (PAULA et al., 2023),

where the individuals in the video have to be predicted and their bounding boxes collected in a pre-processing step. Another example is the Jigsaw puzzles-based pretext task (WANG et al., 2022; SHI et al., 2023; BARBALAU et al., 2023), where some frames (2D) or a set of frames (3D) are divided into patches or tubelets, respectively, and they are shuffled or rotated. Afterward, the model has to predict the correct shuffle position or rotation angle to restore the original order. Another example of such a pretext task is frame ordering, where a model predicts if a frame or a set of frames belongs to a specified video.

3.2.2 Semi-Supervised Video Anomaly Detection

The concept of Semi-Supervised Learning in Video Anomaly Detection (WU et al., 2024), discussed in this section, diverges from traditional Semi-Supervised Learning in general machine learning. The Semi-Supervised concept in VAD refers to the use of only normal videos during the training phase, in which the model has to learn the distribution of the normal videos and detect as anomalies any videos that deviate from this normal pattern.

3.2.2.1 One-Class Learning

This category of VAD aims to learn the normal pattern using only normal examples. Unlike SSL methods, it does not depend on a pretext task and fine-tuning. The different types of methods are described below.

- **One-Class Classifier:** These methods are based on variations of the Support Vector Machine (SVM) (HEARST et al., 1998) for one-class classification, as exemplified by the One-Class SVM (OC-SVM) (SCHÖLKOPF et al., 2001), which aims to estimate a hyperplane that separates all the normal samples on one side. Other variations include SVDD (Support Vector Data Description) and GODS (Generalized one-class discriminative subspaces) (TAX; DUIN, 2004; WANG; CHERIAN, 2019). There are also works that first extract video representations using an autoencoder, and then use these representations as input to OC-SVM (XU et al., 2015). Another variation is SVDD, where instead of estimating a hyperplane, it estimates a hypersphere where it is assumed the normal samples reside (RUFF et al., 2018; LIZNERSKI et al., 2020);

- **Gaussian Classifier:** This approach assumes that normal samples follow a Gaussian distribution, so these methods aim to estimate this distribution using various deep learning

techniques during the training phase (SABOKROU et al., 2017b; FAN et al., 2020a; SABOKROU et al., 2017a). In the test phase, a clip is considered anomalous if it diverges from the trained Gaussian;

- **Adversarial Classifier:** This method aims to use the Discriminator and Generator of a Generative Adversarial Network (see Section 2.1.2.1) as a One-Class Learning approach (SABOKROU et al., 2018b; SABOKROU et al., 2018c). Other works improve this approach, such as Zaheer et al. (2020), which introduces pseudo-anomaly examples.

3.2.2.2 Interpretable Learning

Interpreting the results of a Deep Neural Network is challenging, and it may be important in sensitive application domains like public security. However, Deep Learning-based approaches, like those described in the sections above, lack interpretability. Still, there are works that employ specific strategies to extract insights and explain why an anomaly occurs. For example, Hinami, Mei and Satoh (2017) trains a model using only normal samples, where anomalies are detected through outlier detection. They estimate the normal distribution for the feature vector of people in the scene and for a discrete set of variables describing some attributes of these individuals, such as object type (person, dog, etc.), object action (walking, running, etc.), and object attributes (white, young, etc.). In the test phase, if a person is detected as anomalous using their feature vector, the method provides an explanation based on the discrete set of attributes, with the variable exhibiting the greatest error being the attribute that likely contributes to the anomaly score.

The work of Doshi and Yilmaz (2023) uses a graph to represent the scene, with detected objects as nodes and the relationships between objects as edges. They didn't use self-supervised learning and didn't evaluate on video anomaly detection. There is a discrete set of possible relationships between two objects, given by the Visual Relationship Detection dataset (LU et al., 2016). The benefit of such a representation is the ability to describe an anomaly using discrete values of the relationships, as demonstrated by (HINAMI; MEI; SATOH, 2017). On the other hand, the work of (SINGH; JONES; LEARNED-MILLER, 2023) divides scenes into smaller units called video volumes, which represent specific temporal and spatial segments of the video. It extracts high-level features from these volumes, such as motion direction and speed. A representative subset of these features, known as exemplars, is selected to create a location-dependent model of nominal data. During anomaly detection, test video volumes are compared against these exemplars, with significant deviations classified as anomalies. This

approach enables effective analysis and understanding of scene dynamics for detecting unusual activities.

The AnomalyRuler framework (YANG et al., 2024) introduces a novel rule-based reasoning approach for Video Anomaly Detection (VAD) using Large Language Models (LLMs), marking it as the first reasoning method for one-class VAD. It employs a few-normal-shot prompting technique, allowing for effective anomaly detection with minimal training data, thus avoiding the costly full-shot training. Additionally, AnomalyRuler enhances robustness and performance through strategies like rule aggregation and perception smoothing. Extensive experiments show it achieves state-of-the-art detection accuracy across multiple VAD benchmarks. Overall, AnomalyRuler significantly advances the efficiency and interpretability of VAD.

3.2.3 Multiple Instance Learning (MIL) Approach

Each video is called a bag and is divided into several segments. In the training phase, a deep neural network (DNN)–based model receives the following two input videos: (i) a video containing at least 1 anomaly segment, and (ii) a video containing no anomaly segments. The output of the DNN is an abnormality score for each segment of the two bags (videos). The objective function aims to increase the distance from the higher–scoring segment in the positive bag to the higher–scoring segment in the negative bag, decreasing the number of false negatives and increasing the number of true positives. The videos need to be weakly labelled, which requires much less effort than normal labelling. Thus, this approach uses information from abnormal videos in the training process, unlike the reconstruction error approach, providing an increase in the performance (SULTANI; CHEN; SHAH, 2018).

3.2.4 Weakly Supervised Video Anomaly Detection

In the Weakly Supervised VAD, the samples in the training phase are weakly labeled, which means there is no frame-level annotation in the training set, just a video-level annotation y. Thus, it is not known where the anomaly is in a video, just whether a video has ($y = 1$) or not ($y = 0$) an anomaly. This type of annotation is much cheaper to obtain than frame-level annotation since there is just one label for the video, and moreover, it brings anomaly videos into the training process, unlike the Semi-Supervised VAD that uses only normal videos. Sections 3.2.4.1, 3.2.4.2 and 3.2.4.3 will discuss three important Weakly Supervised Video Anomaly Detection methods.

3.2.4.1 Robust Temporal Feature Magnitude Learning (RTFM) (TIAN et al., 2021)

Consider a set of weakly labelled training videos $\mathfrak{D} = (F_i, y_i)_{i=1}^{|\mathfrak{D}|}$ in which F_i is a matrix $F \in \mathcal{F} \subset \mathbb{R}^{T \times D}$ containing the segment feature vectors extracted by some CNN (i.e., I3D (WANG et al., 2018)), where T is the number of segments into which a video is split (all videos are split into 30 segments) and D is the dimension of the feature vector. Each F_i has a label $y_i \in \{0, 1\}$, where $y_i = 0$ indicates the presence of only normal segments in F_i, and $y_i = 1$ indicates the presence of at least one anomaly segment in F_i. Given a set of video segments F, the RTFM method predicts whether each segment is anomalous or not using the model $f_\phi(s_\theta(F))$, where $s_\theta : \mathcal{F} \rightarrow \chi$ is a temporal feature extractor, with $\chi \in \mathbb{R}^{T \times D}$, and $f_\phi : \chi \rightarrow [0, 1]^T$ is the segment classifier. The loss function used to minimise the parameters θ and ϕ is (TIAN et al., 2021)

$$\min_{\theta,\phi} \sum_{i,j=1}^{|D|} l_s(s_\theta(F_i), s_\theta(F_j), y_i, y_j) + l_f(f_\phi(s_\theta(F_i)), y_i), \tag{3.1}$$

where s_θ is implemented using a pyramid of dilated convolutions (YU; KOLTUN, 2015) over the time domain to capture dependencies from neighbouring segments. Additionally, a temporal self–attention module (WANG et al., 2018) is used to help aggregate information over all the video segments. f_ϕ is a fully connected neural network with 3 layers that determines whether the segments represent an anomaly or not. l_f is a loss function to train the segment classifier, and l_s is a loss function that aims to minimise the k largest segment feature magnitudes from the normal videos and maximise the k largest segment feature magnitudes from abnormal videos (TIAN et al., 2021):

$$l_s(s_\theta(F_i), s_\theta(F_j), y_i, y_j) = \begin{cases} max(0, m - d_{\theta,k}(X_i, X_j)), & \text{if } y_i = 1,\ y_j = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{3.2}$$

where m is a predefined margin and $X_i = s_\theta(F_i)$ is the abnormal temporal feature extracted from video i (X_j is similarly defined). $d_{\theta,k}(X_i, X_j))$ is given by (TIAN et al., 2021)

$$d_{\theta,k}(X^+, X^-)) = g_{\theta,k}(X^+) - g_{\theta,k}(X^-), \tag{3.3}$$

and $g_{\theta,k}$ is given by (TIAN et al., 2021)

$$g_{\theta,k}(X) = \max_{\Omega_k(X) \subseteq \{x_t\}_{t=1}^T} \frac{1}{k} \sum_{x_t \in \Omega_k(X)} ||x_t||_2, \tag{3.4}$$

where the temporal feature vectors X can also be denoted by $X = (x_1, x_2, \ldots, x_T)$, and $\Omega_k(X)$ represents the set containing k temporal feature vectors from X, so $|\Omega_k(X)| = k$. The main assumption of this work is that the video segment feature vectors can be classified using their magnitudes (ℓ_2 norm) instead of their spatial separation. This method assumes that the feature vector magnitudes from the normal video are smaller than those from the abnormal videos. Thus, in Equation (3.4), the mean of the k largest feature vector magnitudes from video X is calculated; the subtraction of Equation (3.3) guarantees that the mean magnitude of the abnormal videos (X^+) is as large as possible, while the mean magnitude of the normal videos (X^-) is as small as possible. θ in Equations (3.3) and (3.4) indicates that in the training process, the feature extractor s_θ will be optimised to produce feature vectors that satisfy this main assumption.

3.2.4.2 Weakly Supervised Anomaly Localisation (WSAL) (LV et al., 2021)

Still considering what was defined in the previous subsection, in a set of weakly labelled training videos $\mathfrak{D} = (F_i, y_i)_{i=1}^{|\mathfrak{D}|}$, each video segment in F has a feature vector extracted from some CNN. These feature vectors can be denoted by $F = (f_1, f_2, \ldots, f_T)$. The WSAL method uses Equation (3.5) to define whether each segment is anomalous or not (LV et al., 2021):

$$S(F) = \max_{j,k=1,\ldots,T} g(\psi(f_{j-d}, .., f_j, \ldots, f_{j+d}), \psi(f_{k-d}, .., f_k, \ldots, f_{k+d})), \tag{3.5}$$

where ψ is a DNN–based architecture called high–order context encoding that aims to extract temporal and spatial information from video frames; it receives an anchored video segment f and its $2d$ neighbours to detect temporal features in a window $[-d, d]$. g is a function that measures the anomaly score margin between the segments f_j and f_k. The max function is used to obtain the two segments with a greater difference in terms of the abnormality score, taking into account the whole video (i.e., all values of j and k). For the normal videos, it is intended that all segments should present a ψ value near 0, resulting in a g value near 0. In the abnormal videos, it is intended that the g value should be higher because the anomalous segment will result in a high ψ value. Thus, the result of $S(F)$ is intended to be high when F is anomalous ($y = 1$) and low when F is normal ($y = 0$). Thereby, one can understand $S(F)$ as a function that computes the anomalous score of a video F. Thus, the base loss function used to guide the optimisation is given by (LV et al., 2021)

$$\Im(F) = max\{0, 1 - \frac{1}{n_1}\sum_{i=1}^{n}[S(F)|y = 1] + \frac{1}{n_0}\sum_{j=0}^{n}[S(F)|y = 0]\}, \tag{3.6}$$

where n_1 and n_0 represent the total number of anomaly samples and the total number of normal samples, respectively.

The high–order context encoding (HCE) module uses two components to mine variations from video features: (i) the semantic component and (ii) the variation component. For the semantic component, it is assumed that the videos can be interpreted as a time series, and the extraction of high–level features can be performed as a regression task. In this way, given feature vectors $(f_1, f_2, \ldots, f_T)$ extracted from consecutive segments, the regression approach can be formulated as follows (LV et al., 2021):

$$\tilde{f}_t = \sum_{j=-d,\ldots,d,d\neq 0} W_d f_{t+d} + W_0 f_t + b, \tag{3.7}$$

where W_d is a projection function in the d–th segment and b is a bias term. The idea is to obtain discriminative information using a window of $2d$ segments related to the t–th segment, capturing immediate spatial semantics and local dynamic variations (TIAN et al., 2021). Then, the WSAL method uses a fully connected network to obtain the final semantic anomaly score for segment t, which can be expressed as follows (LV et al., 2021):

$$\psi^{sem}(\tilde{f}_t) = \sigma(w_{sem}\tilde{f}_t + b_{sem}), \tag{3.8}$$

where w_{sem} and b_{sem} are the weights and biases of the fully connected network. The $\psi^{sem}(\tilde{f}_t)$ network can be stacked so that it has several layers, but it was found that just one layer performed well.

The variation component pays attention to the variation from two adjacent segments, opposing the semantic component, which pays attention to the variation from a window of d segments. To do this, the variation component uses the metric $1 - cos(\tilde{f}_{t-1}, \tilde{f}_t)$, which results in large values for dramatic changes in segments, where $cos(\tilde{f}_{t-1}, \tilde{f}_t)$ is given by (LV et al., 2021)

$$cos(\tilde{f}_{t-1}, \tilde{f}_t) = \tilde{f}_{t-1}^{\top}\tilde{f}_t / (\left\|\tilde{f}_{t-1}\right\|^2 \left\|\tilde{f}_t\right\|^2). \tag{3.9}$$

Then, the following second–order discrepancy is calculated to obtain the anomaly score (LV et al., 2021):

$$\psi^{var}(\tilde{f}_t) = (2 - cos(\tilde{f}_{t-1}, \tilde{f}_t) - cos(\tilde{f}_t, \tilde{f}_{t+1}))/4, \tag{3.10}$$

where the division by four normalises the anomaly score so that it falls within the range [0,1].

The two semantic components and variation components give context cues that can be used to detect abnormalities in video segments. The L1–distance is used to measure the variation from consecutive segments (LV et al., 2021):

$$S^{sem}(F) = \max_{i,j=1,..,T} |\psi^{sem}(\tilde{f}_i)) - \psi^{sem}(\tilde{f}_j))|, \tag{3.11}$$

$$S^{var}(F) = \max_{i,j=1,..,T} |\psi^{var}(\tilde{f}_i)) - \psi^{var}(\tilde{f}_j))|. \tag{3.12}$$

Putting S^{sem} and S^{var} into Equation (3.6) results in two new margin losses, S^{sem} and S^{var}, respectively. Thus, the final loss function, which takes into account the semantic component and variation component, is given by (LV et al., 2021)

$$S^0 = S^{sem}(F) + S^{var}(F) + \frac{\beta}{n}\sum_{i=1}^{n}\sum_{t=1}^{m}(|s_t^{sem}| + s_t^{var}), \tag{3.13}$$

where β is a sparsity constraint, and s_t^{sem} and s_t^{var} are the anomaly scores given by Equations (3.11) and (3.12), respectively. An approach for dealing with simulated noise data for data augmentation was also proposed, but the formulation is not within the scope of this work. Mathematical proofs, theoretical motivations, and further equation derivations can be found in (LV et al., 2021).

3.2.4.3 Real-world Anomaly Detection in Surveillance (RADS) (SULTANI; CHEN; SHAH, 2018)

Given a set of weakly labelled training videos $\mathfrak{D} = (F_i, y_i)_{i=1}^{|\mathfrak{D}|}$, each video segment in F_i has a feature vector extracted from some CNN, given by $F = (f_1, f_2, \ldots, f_T)$. Given a set of features F_a extracted from an anomaly video ($y = 1$) and a set of features F_b extracted from a normal video ($y = 0$), the RADS objective function is given by (SULTANI; CHEN; SHAH, 2018)

$$l(F_a, F_b) = max(0, 1 - \max_{i \in F_a} \xi(f_i) + \max_{i \in F_b} \xi(f_i)), \tag{3.14}$$

where ξ is a 3–layer fully connected network that maps the feature vector f_i to an anomaly score. In this equation, the higher–scoring segment in F_a has to be as close as possible to 0, while the higher–scoring segment in F_b has to be as large as possible, satisfying the assumption of multiple instance learning, which is explained in Section 3.2.3. The main disadvantage of this approach in comparison to RTFM (Section 3.2.4.1) and WSAL (Section 3.2.4.2) is that it does not consider the temporal axis to calculate the anomaly scores of video segments.

3.2.5 Fully Video Anomaly Detection

This category means the VAD training process has frame-level annotation regarding where the anomalies are in the videos. The exact start and end positions of each anomaly in each video are used as supervisory signals. However, different feature extractors can be applied to the raw pixel frames to acquire high-level features for the VAD method. They are described below.

- **Raw pixels:** The first approach to process the videos is using the raw pixels extracted from the video frames. This approach can improve the performance of anomalies that are better represented from a visual perspective (WU et al., 2024). This category also comprises methods that use visual embeddings extracted from frames using some feature extractor, like Convolutional Networks or ViT (Vision in Transformer). Examples of works that follow this line are Dong, Qin and Wang (2016), Zhou et al. (2017), Perez, Kot and Rocha (2019);

- **Motion:** The works comprised in this category aim to make a pre-processing step in the raw frames to extract the motion of the objects between the frames and use it to find the anomalies. Such an approach can be useful in cases where the anomalies are better identified through their motion rather than their appearance. Examples of works in this area are Dong, Qin and Wang (2016), Hanson et al. (2018), Sudhakaran and Lanz (2017);

- **Skeleton:** Anomalies based on human actions, like fights or robbery, can be better represented by the human pose. In this way, a pre-processing step can extract the human pose, known as a skeleton, which is a set of coordinates representing points of the human body. One advantage is that such representation can improve the detection of these human-based anomalies. Another advantage is that the computational power required to process only a few coordinates is much lower than processing raw pixels. However, the disadvantage is that a tracking algorithm has to be executed in the pre-processing step to estimate the human in the video, which can cause an increase in the deployment cost of such a method. Examples of methods include Su et al. (2020), Singh, Patil and Omkar (2018);

- **Audio:** Audio can represent important additional information for a VAD method across a range of anomaly types. The types in this category use audio information alone or together with other features in the training and inference process. One can highlight the method Peixoto et al. (2020) that uses audio as part of VAD detection;

- **Hybrid:** The categories described above can be used together. The methods in this category are called hybrid. Examples of such methods are Cheng, Cai and Li (2020) that use raw frames together with frame motion, Garcia-Cobo and SanMiguel (2023) that use skeleton and frame motion, and Shang, Wu and Liu (2022) that combines raw frames with audio.

3.2.6 Unsupervised Video Anomaly Detection

This category encompasses methods designed to detect anomalies in videos without relying on labeled training data.

- **Pseudo label-based paradigm:** The key aspect of this paradigm is to use iterative refinement of a model using pseudo labels generated by another approach. For example, Wang et al. (2018) estimates anomaly labels from unlabeled samples using an auto-encoder trained with a reconstruction loss. These estimated labels are then used on a One-Class Support Vector Machine (OC-SVM) to detect anomalies and normal events (PANG et al., 2020). Note that, in this work, first the labels are estimated using a One-Class method, described in Section 3.2.1. Other authors use different methods to estimate the labels, but the Pseudo label based paradigm focuses on the iterative refinement. Other works use different methodologies to generate pseudo labels in an unsupervised way, as in Zaheer et al. (2022), where they use a generator and a discriminator, one generating pseudo labels for the other;

- **Change detection-based paradigm:** This paradigm aims to classify video segments using the change in the temporal axis. The work of Hu et al. (2022) tries to detect persons and objects in the scene, acquire the spatio-temporal tubelet, and mask and reconstruct some parts of it using a Transformer. Using a thresholding technique, an anomaly or normal event is detected. Note that this is different from One-Class Learning (Section 3.2.1) because the training set can contain both normal and anomalous data.

Table 4 compares some AUC (Area Under the Curve) ROC (Receiver Operating Characteristic) performance of multiple methods for multiple datasets.

Table 4 – Frame-Level reported for several Video Surveillance Anomaly Detection methods for several datasets. If a methods has several configurations, the selected configuration is the one with better results in UCF-Crimes.

Refs.	ShagayTech	USPed1	USPed2	Avenue	UMN (Average over scenes)	LV Dataset	UCF-Crimes	Sub-Entrance	Sub-Exit
	AUC								
Lv et al. (2021)							85.4		
Tian et al. (2021)	97.2						84.3		
Sultani, Chen and Shah (2018)							75.4		
Khan, Park and Kyung (2018)		-	-		95.2	23.9	64.3		
Chen et al. (2020)		87.2	96.5	87.4					
Rezaei and Yazdi (2021)		93.9	97.3						
Ramchandran and Sangaiah (2020)		98.4	95.7	90.7					
Doshi and Yilmaz (2020)	71.6	97.8		86.4					
Duman and Erdem (2019)		92.4	92.9	89.5					
Mondal and Chanda (2018)		79.0	94.0		98.0				
Fan et al. (2020b)		97.1	99.4		99.6				
Sun, Shao and He (2019)		91.4	91.1						
Chu et al. (2018)		90.9	90.2	82.1					
Ma, Wang and Yuan (2014)					99.3				
Wang et al. (2020)		65.2			98.1				
Amraee et al. (2018)		68.0	85.0						
Feng, Liang and Li (2021)			84.5	80.3				87.3	90.8
Wang et al. (2019)		89.7	89.6		99.7				
Yang, Liu and Wu (2021)	74.5	84.7	97.6	88.6					
Wu et al. (2020)		84.1	92.4						
Xu, Sun and Jiang (2019)		95.1	95.3						
Wang, Yao and Yao (2021)	75.4	85.3	96.3	86.7					

4 CamNuvem: A robbery dataset for video anomaly detection

This Section going to guide the reader through the our proposed CamNuvem dataset, containing 486 new real–world video surveillance clips showing robberies that were acquired from public sources such as social media is presented. Section 4.1 discusses the statistics of the dataset, and Section 4.2 details the construction of the initial benchmark involving three video surveillance anomaly detection methods.

4.1 CamNuvem Dataset Statistics

The CamNuvem dataset consists of 972 videos in total, including 486 abnormal videos collected from public sources such as YouTube, Facebook, and other social media websites. The videos were edited to remove useless segments that were not recorded by the surveillance camera, such as introductions and endings. All the abnormal videos consist of people committing robberies. This category of videos was split into 437 videos for the training set and 49 videos (10%) for the test set. As with UCF–Crime (SULTANI; CHEN; SHAH, 2018), this dataset is weakly labelled, which means that only video–level labels are available for the training videos and frame–level labels are provided for the test videos. The annotators marked a video segment as abnormal only if a robbery was explicitly happening. If a hypothetical camera captures people committing a robbery, and then these people exit the camera view field and return moments later, there will be two abnormal segments with a normal segment between them. The decision to make a segment abnormal does not depend on past events. This feature can be inserted in future datasets and explored in future studies.

There are also 486 normal videos extracted randomly from UCF–Crime; 437 of these videos were used for the training set and 49 were used in the test set. Thus, the main novelty of this dataset is the aggregation of a new robust set of abnormal videos to improve the development of robbery detectors, which can be used with datasets that are already available.

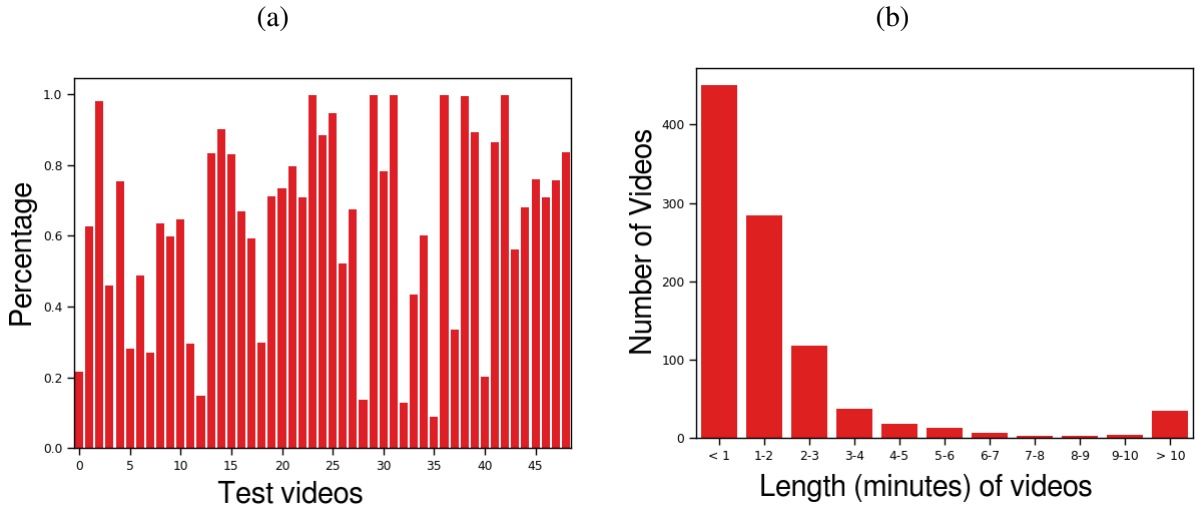
All the videos were adjusted to 320×240 px and 30 fps. Figure 13a shows the anomaly proportion in each test video, i.e., the percentage of anomaly frames in each video. As described

previously, the training videos have no frame–level annotations. Figure 13b shows a video–duration histogram of the dataset. Finally, Figures 14 and 15 show some frame samples in the abnormal set to present the general format of a robbery. Note that even though this dataset is focused on only one type of anomaly (robbery), this anomaly is variable enough to make it hard to collect and label a representative dataset in order to train a fully supervised robbery detector.

As shown in Figure 14, in the real world, surveillance cameras can provide different points of view, which is evident when one compares the camera position in the sample shown in Figure 14a–d to the camera position in the sample shown in Figure 14e–h. The camera in the former sample captured the images at a lower height than the camera in the latter sample. Each sample in the dataset was constructed by a camera fixed in a specific position, which makes the dataset representative of the real world.

Furthermore, as shown in Figure 15, there is a large number of different features that can indicate a robbery, e.g., (i) the man with an orange shirt who is lying down in Figure 15c,d, (ii) the interaction of the robber and the victim in Figure 15g,h, (iii) the man with his hands in the air in Figure 15e, and (iv) the man wielding a gun in Figure 14a. One can conclude that the features that compose a robbery situation have a high variability and may also have a large temporal window.

Figure 13 – CamNuvem dataset statistics. (**a**) Anomaly percentage of each test video. (**b**) The number of videos in the dataset.



Source: Author's own work.

4.2 Initial Benchmark

The values of the area under the curve (AUC) from the receiver operating characteristic (ROC) curve for the CamNuvem dataset were obtained using the video anomaly detection methods RTFM (TIAN et al., 2021), WSAL (LV et al., 2021), and RADS (SULTANI; CHEN; SHAH, 2018). These methods were chosen due to their high performance in terms of the AUC when they were applied to the UCF–Crime dataset. These techniques expect pre–extracted video features, which must be obtained by applying some feature extraction method to the raw video frames. There are various methods that can be used for this task, but in this work, I3D was used (WANG et al., 2018). This design choice is explained below.

Traditional convolutional blocks, when applied to a single image, can aggregate features due to their hierarchical nature. Thus, a CNN can predict some value in the final layer by taking into account features and objects that are in the same image. However, a video may be seen as a set of images (frames) that are linked together in a well-defined order; i.e., each frame is linked to the posterior and previous frames, and information flows through these frames along the temporal axis. When a CNN is applied to a video dataset to extract information, the traditional convolution block can see only a single frame; it ignores the features in the neighbouring frames, and, therefore, on the temporal axis, which can represent an important information loss.

Figure 14 – Two samples with clearly different view fields are shown. Four frames from two different videos are illustrated here. The camera used to capture the images from the first video (**a**–**d**) is at a greater height and the people were captured from above. The camera used to capture the images from the second video (**e**–**h**) is at a lower height and the people were captured from the front.

(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



Source: Author's own work.

Figure 15 – Samples in the abnormal category. Four frames from two different videos are illustrated here. Several image elements can be used to detect a robbery. In the first video (**a**–**d**), the robbery is indicated by the man crouching in the first (**a**) and last (**d**) frames. In the second video (**e**–**h**), the robbery is indicated by the man raising his hands in the first frame (**e**).

(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)



Source: Author's own work.

A natural extension that allows the traditional convolution block to deal with temporal datasets (such as video datasets) involves adding a dimension to the convolutional kernels. This technique is known as convolution 3D (C3D) (TRAN et al., 2015). In this technique, a single convolutional kernel will output a weighted sum of the pixels of a neighbouring frame set, while traditional convolution outputs a weighted sum of the pixels of only one frame, which represents only one moment on the temporal axis. Thus, C3D can be thought of as a convolutional block that takes into account features and information across the temporal axis (when dealing with video datasets).

However, convolution 3D aims to acquire local and temporal features from datasets, and long–range features are aggregated together in the hierarchical neural network structure. This hierarchical structure has a negative impact on computational optimisation because the signals captured by single kernels represent information from a small patch of the image, and the only way to capture long–range dependencies between features is to feed this signal to the deepest layers repeatedly, causing computational inefficiency due to parallelisation constraints. This scenario also occurs in recurrent neural networks (RNNs), which are often utilised for textual datasets; long–range features have to be repeatedly passed through the forward layers to be aggregated together, causing computational difficulties and limiting the power of an artificial neural network.

To confront this problem, non–local neural networks (WANG et al., 2018), which were inspired by the classical non–local means operator (BUADES; COLL; MOREL, 2005), were proposed; they include a neural network block that can improve the computational parallelisation. In this method, the response of a non–local block is a weighted sum of all positions in the input feature map, unlike in convolutional blocks, where the response is computed as a weighted sum of a small local patch of the input feature map. Thus, the video anomaly detection methods (RTFM, WSAL, and RADS) used in this work receive pre–extracted features from raw videos that were extracted by the I3D ResNet Nonlocal model (I3D) (WANG et al., 2018).

4.2.1 Results

The WSAL (LV et al., 2021), RTFM (TIAN et al., 2021), and RADS (SULTANI; CHEN; SHAH, 2018) methods were used as a baseline for the new dataset introduced in this work. The methods have been previously tested on the UCF–Crime dataset, but in the official RTFM implementation, the four 224×224 px corners and central crop, plus their horizontal flipped

versions, were extracted from each video frame, resulting in 10 independent images. In the training and test phases, these 10 images are given to the RTFM anomaly detection model and 10 individual anomaly scores are acquired. However, the mean of these 10 independent anomaly scores is calculated to determine whether the original video is an anomaly video or not. Note that this procedure is different from the traditional data augmentation approach, which would result in 10 individual error corrections in the training phase. This procedure will be called 10–crop in the remainder of this work. The original implementations of WSAL and RADS do not perform 10–crop, making a fair comparison of them difficult. Thus, this work created new versions of WSAL and RADS to allow 10–crop samples to be used as input, following the same procedure as RTFM. Likewise, this work also implemented a new version of RTFM to allow non–10–crop samples to be used as input. Thus, a fair comparison of the results of these methods can be performed. The CamNuvem dataset was used in these experiments.

As explained previously, the features were extracted using the I3D model, which was pre–trained on the Kinetics dataset (CARREIRA et al., 2019). To verify the influence of the 10–crop method, two versions of the video features were extracted: (i) the first was extracted using the entire video frame, and (ii) the second was extracted using the 10–crop methodology. To avoid frame redundancy, the feature vector was created using sets of 16 consecutive frames, and each feature vector had a size of 2048. All the experiments were performed on a machine with 32 GB of memory and an NVIDIA 1660 Super graphical processing unit with 6 GB of memory.

In Table 5, the values of the AUC ROC obtained for the RTFM, WSAL, and RADS methods using the CamNuvem dataset with the I3D feature extractor are presented. The values are divided into ‘No 10–crop’ and ‘10–crop’ categories. Moreover, for the CamNuvem dataset (Table 5), the analysis was performed using the entire test partition of the dataset (entire test set row) and using only the abnormal videos in the test partition (only the abnormal videos in the test set row). This division was made because even in an anomaly video, there are normal frames, and because there are many more normal video frames than abnormal video frames in the dataset, even a bad classifier can appear to perform well if it classifies every segment as normal; this can obscure the true performance of the classifier when only the AUC ROC value is considered. To improve the understanding of the capabilities of these methods, the AUC ROC values were also evaluated for only the abnormal videos in the dataset. These values can be seen in the last row of Table 5.

Table 5 – AUC ROC values of the anomaly detection methods RTFM, WSAL, and RADS, which were used to detect abnormalities in the CamNuvem dataset. The features were extracted by I3D.

RTFM		**WSAL**		**RADS**	
No 10–crop	**10–crop**	**No 10–crop**	**10–crop**	**No 10–crop**	**10–crop**
Entire test set					
78.76%	88.75%	82.01%	83.45%	79.08%	82.48%
Only the abnormal videos in the test set					
56.03%	66.35%	55.22%	54.59%	51.68%	50.51%

Note that in Table 5, considering the entire test set , the RTFM method obtained an AUC value of 78.76% for the 'no 10–crop' dataset and an AUC value of 88.75% for the 10–crop dataset. These values suggest that the 10–crop data augmentation methodology increases the quality of the classification. However, the AUC values decrease for only the abnormal videos in the test set in Table 5; the AUC values decrease to 56.03% and 66.65% for the 'no 10–crop' and 10–crop datasets, respectively. These AUC values decrease by 22.73% and 19.40%, respectively. These values suggest that the RTFM method does not perform well when it has to localise the anomaly in an abnormal video. For the WSAL results shown in Table 5, this pattern repeats: the AUC values for the entire test set were 82.01% and 83.45% for the 'no 10–crop' and 10–crop datasets, respectively, and these values drop for only the abnormal videos in the test set to 55.22% and 54.59%, decreasing by 26.79% and 28.86%, respectively. Note that the increase in the AUC value for the 10–crop dataset was only 1.44%, unlike the RTFM method, which achieved an improvement of 9.99%. In the RADS results shown in Table 5, the pattern is similar; the AUC values for the entire test set were 79.08% and 82.48% for the 'no 10–crop' and 10–crop datasets, respectively, and these values drop for only the abnormal videos in the test set to 51.68% and 50.51%, decreasing by 27.4% and 31.97%, respectively.

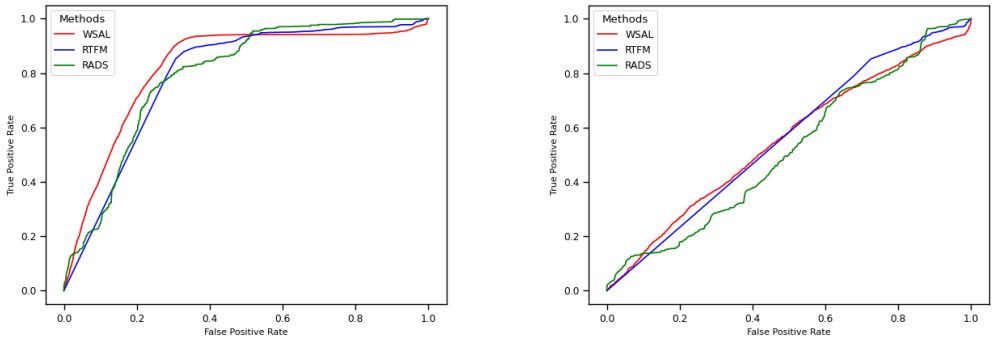
The values shown in Table 5 are illustrated in Figures 16 and 17. Figure 16 depicts the RTFM (blue line), WSAL (red line), and RADS (green line) ROC curves for the entire test set (Figure 16a) and for only the abnormal videos in the test set (Figure 16b) for the 'no 10–crop' version of the dataset. Note in Figure 16b that the three methods perform almost randomly when they are evaluated using only anomaly videos, which indicates that further research into robbery anomaly detection and localisation using surveillance videos is required. Likewise, Figure 17 depicts the RTFM (blue line), WSAL (red line), and RADS (green line) ROC curves for the entire test set (Figure 17a) and for only the abnormal videos in the test set (Figure 17b) for the 10–crop version of the dataset. Note that in Figure 17b, the three methods perform almost randomly, but

the RTFM method (blue line) achieves a subtle improvement in the performance.

In Figures 16b and 17b, the decrease in the AUC value when the methods are evaluated only on the anomaly videos is due to their inability to localise the anomalies in the videos. When the entire test set is used (Figures 16a and 17a), what is being measured is the ability of the method to identify whether or not a video contains an anomaly. When only the anomaly videos are used, what is being measured is how well the methods can predict where the anomalies are in the video. Thus, what the experiments show is that all of the anomaly detector methods performed poorly when it came to finding the exact positions of the anomalies.

Taking into account the analysis of the entire test set and only the abnormal videos in the test set, the AUC values improved when the 10–crop version of the dataset was used. However, each crop is considered an individual image, so this version requires more computational power.

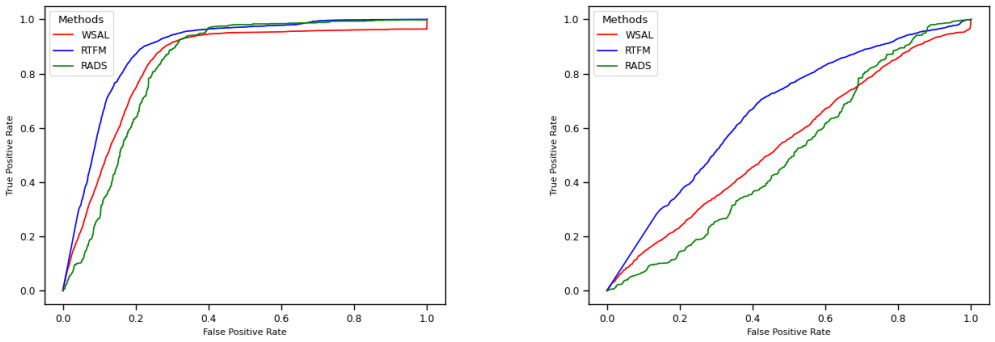
Figure 16 – RTFM, WSAL, and RADS ROC curves without 10–crop for CamNuvem dataset. (**a**) Entire test set. (**b**) Only the anomaly videos.



(**a**) (**b**)

Source: Author's own work.

Figure 17 – RTFM, WSAL, and RADS ROC curves with 10–crop for CamNuvem dataset. (**a**) Entire test set. (**b**) Only the anomaly videos.



(**a**) (**b**)

Source: Author's own work.

4.3 Conclusions

This Chapter presented the CamNuvem dataset, a new weakly labelled dataset containing real–world surveillance videos showing robberies. The previous sections discussed the importance and innovation of this dataset, which will enable the development of human behaviour detection systems that can be used to detect robberies in real life. An initial benchmark was also constructed to allow a comparison of the results with future studies.

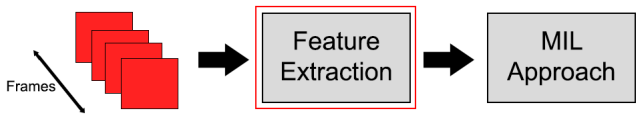
The initial benchmark demonstrated that the feature extraction technique strongly influences the results. Furthermore, it was also shown that the methods are not good at localising anomalies in videos.

5 Self-Supervised Feature Extraction for Video Surveillance Anomaly Detection

The main objective of this Chapter is to presents a feature extraction methodology designed to leverage videos that involve interactions between objects and/or humans, aiming to capture the relationships between then in a scene. To achieve this objective, the Self-Supervised Learning approach provides an ideal framework. It involves (a) a pretext task, where a Deep Learning Network (DNN)-based model known as a pretext model is trained in an unsupervised manner, and (b) a downstream task, where the trained pretext model is fine-tuned for the desired task, such as video surveillance anomaly detection.

It is important to note that this Chapter focuses only on the feature extraction step and does not propose a MIL approach. Instead, it employs existing video surveillance anomaly detection approaches to evaluate the video features extracted by SSOC on a weakly labeled video anomaly detection dataset. Figure 18 illustrates the location of this work's proposal within the video surveillance anomaly detection pipeline.

Figure 18 – The SSOC approach proposed in this paper focuses on the feature extraction step (represented by the red rectangle) in the video surveillance anomaly detection pipeline. To obtain the results, existing MIL methods are employed.



Source: Author's own work.

The code for this work is available publicly at <https://github.com/daviduarte/ssoc>. Furthermore, an open-source platform was developed in this study to standardize feature extraction and comparison among different MIL-based video surveillance anomaly detection methods. The source-code is also available publicly at <https://github.com/daviduarte/camnuvem-research-platform>. Ensuring a fair comparison has been a significant challenge, given the variations in data preparation methodologies employed by different studies (PAULA; SALVADEO; ARAUJO, 2022).

Section 5.1 will provide details about the SSOC. Section 5.2 will discuss the results achieved. Section 5.3 will present the discussion and conclusions.

5.1 Methodology

The objective of the SSOC approach is to train a DNN-based model that can understanding the relationships between objects and/or humans over time using an unsupervised approach. The goal is to facilitate transfer learning to other video understanding tasks, such as video surveillance anomaly detection. The self-supervised approach is well-suited for achieving this objective because it leverages the inherent structure of the dataset to automatically generate labels that optimize a pretext DNN-based model. This enables the model to understand the dataset's intrinsic structure, thereby allowing fine-tuning in a downstream task.

Section 5.1.1 will provide a description of the pretext task, which focuses on learning the object relationships in the scene. Section 5.1.2 will discuss how the pretext model, which has been optimized in the pretext task, will be fine-tuned using video surveillance anomaly detection datasets.

5.1.1 Pretext Task

In the pretext task, a reference object r is selected within a small video segment, and its position and appearance feature vector F in a future frame are estimated based on a set of past frames, as indicated by Equation 5.1, where p represents the set of past frames and $\mathbf{g}$ represents a DNN-based model:

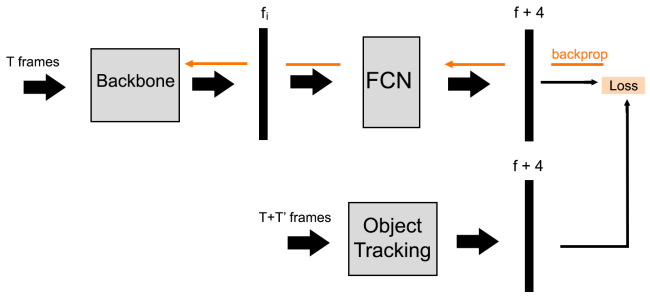
$$\mathbf{g}(p) = F \tag{5.1}$$

By estimating the position and appearance of an object in the future, it is expected that the model will learn how the objects influence each other and their intrinsic relationships. The term "object" here is generic and several object classes could be used, like "person", "car", "motorcycle", etc.

This approach is depicted in Figure 19. A temporal window (or scene) with $T + T'$ frames is considered, where the first T frames are used as the set of past frames p to predict the feature vector F for the reference object r in the $(T + T')$-th future frame. The reference object r is chosen

as the object in the first frame of the temporal window with the highest confidence score. The method uses the Object Tracking module (gray square at the bottom of Figure 19) to provides the re-identification and tracking of object r in any of the T frames. Therefore, given the object r in the first frame and using the Object Tracking module, it is possible to access its position and appearance in the $(T+T')$-th frame (the last frame in the temporal window). This will be the label in the training process. The appearance of an object is represented by a vector $F_a \in \mathbb{R}^f$, where f is the vector size, extracted from an intermediate layer of some object detector (implemented as part of the Object Tracking module), and it is named the appearance feature vector. The position of an object is represented by a vector $X \in \mathbb{R}^4$, named the position feature vector, which describes the bounding box of the object $(x_1, y_1), (x_2, y_2)$, where the first 2D coordinate represents the top-right bounding box corner and the second 2D coordinate represents the bottom-left bounding box corner. Concatenating the appearance feature vector and the position feature vector results in the object feature vector $F \in \mathbb{R}^{f+4}$. The feature vector F of the reference object r is shown in the black rectangle in the output of the Object Tracking in Figure 19.

Figure 19 – The pretext task of the SSOC approach. This tasks aims to teach a DNN-based model how the objects interact in the scene.

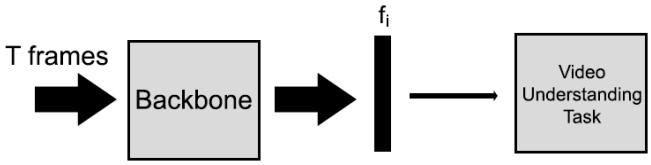


Source: Author's own work.

The pretext model is composed of a deep neural network called backbone model, followed by a fully connected network (FCN) represented on top of Figure 19. The backbone receives the first T frames from the scene as input and outputs a feature vector of size f_i, corresponding to an intermediate layer that serves as a representation for the first T frames of the scene. This feature vector is then passed to the FCN, which aims to estimate the coordinates and appearance F of size $f+4$ for the reference object r in the future $(T+T')$. The real coordinates and appearance F of the reference object r in the $(T+T')$-th frame are estimated by the Object Tracker, which computes it through the re-identification of object r in all the $T+T'$ frames.

In the training process, the output of the FCN and Object Tracker is compared and the

Figure 20 – Downstream task for the SSOC approach.



Source: Author's own work.

error is backpropagated through the FCN and backbone networks. The weights of the Object Tracker are kept frozen during this process. The quality of the training depends on the quality of the implementation of the Object Tracker and the object detector implemented in it to perform re-identification in the frames. The idea is to force the backbone model to predict the main object of the first frame of the scene (the one with the highest score in the Object Tracking's object detector) in the future. Note that, as a design consideration, the backbone model does not explicitly detect the objects in the scene to keep the computational requirements low. In this work, the Object Tracker module is implemented using the YoloV5 (KHANAM; HUSSAIN, 2024) as object detector.

5.1.2 Downstream Task

In the downstream training phase, only the backbone network from the pretext model is used. The idea here is that the backbone weights have been modified in the pretext task to understand the object relations in the scene. The FCN is excluded following the transfer learning methodology, where the last layers are removed to utilize only the features generated from the backbone. In this task, the resulting model, shown in Figure 20, is called the downstream model, and the training process will fine-tune this model.

Although this work focuses on fine-tuning in the video surveillance anomaly detection task (detailed in Section 5.1.3), this methodology can be applied to any video dataset that presents relations between different types of objects. Therefore, Figure 20 contains a generic block labeled as "Video Understanding Task" (last gray box) to represent the downstream task.

In the training process, the error can be backpropagated through the backbone network, or new trainable layers can be inserted while keeping the backbone weights frozen. This choice depends on the experimental setup. Details of the experimental setup are described in Section 5.1.3.

5.1.3 Experimental Setup

As explained earlier, the video understanding task used to measure the proposed approach is video surveillance anomaly detection using Multiple Instance Learning (MIL) (SULTANI; CHEN; SHAH, 2018). Among the many approaches available, the Robust Temporal Feature Magnitude learning (RTFM) (TIAN et al., 2021), Weakly Supervised Anomaly Localization (WSAL) (LV et al., 2021), and Real-World Anomaly Detection Surveillance (RADS) (SULTANI; CHEN; SHAH, 2018) were chosen. RTFM and WSAL are modern techniques that are top-ranked on the UCF-Crime dataset, while RADS is a traditional technique commonly used in the evaluation of MIL works (TIAN et al., 2021; LV et al., 2021). Therefore, to clarify, the RTFM, WSAL, and RADS methods were used in the Video Understanding Task block shown in Figure 20.

The datasets used to generate the results are UCF-Crime (SULTANI; CHEN; SHAH, 2018) and CamNuvem (PAULA; SALVADEO; ARAUJO, 2022), sampled at 30 fps (original sampling rate). The parameters T and T' are set as $T = 16$ and $T' = 16$, so the objective is to teach the backbone model to predict the feature vector F of the reference object r 16 frames into the future, given the 16 past frames. These values were set empirically. We chose a small window initially to demonstrate the effectiveness of the SSOC method.

In the pretext task, the training phase used the training partition, and the test phase used the test partition of both the CamNuvem and UCF-Crime datasets. In the downstream task, both the training and test phases also utilized the training and test partitions of the CamNuvem and UCF-Crime datasets. There was no mixing of samples from these datasets. At no point were test samples used to adjust weights in the training phase, whether in the pretext task or the downstream task. The feature vector f_i (the output of the Backbone in Figure 19) has a size of 2048, while the feature vector $f + 4$ (the output of the FCN) has a size of $1280 + 4 = 1284$. The loss function used was Mean Squared Error (MSE).

Regarding the transfer learning methodology, this work chooses to keep the weights of the backbone model frozen in the downstream task (Figure 20) and add new trainable layers on top of the downstream model. These layers are part of the RTFM, WSAL, and RADS architectures.

In the training of the downstream task, we followed the experimental methodology of RTFM, WSAL, and RADS, which means that all the final feature vectors of size $f + 4$ generated from the first T frames of the temporal windows were averaged to result in 32 segments. RTFM

was trained for 1500 epochs with a learning rate of 0.001, using normalized input data. WSAL was trained for 500 epochs with a learning rate of 0.0001 for I3D and 0.001 for SSOC+I3D, as the latter did not converge with the former learning rate value. The input data was standardized for WSAL. RADS was trained for 75 epochs with a learning rate of 0.001, using normalized data. The utilized loss function was MSE.

The I3D model (CARREIRA; ZISSERMAN, 2017) was used as the backbone model because it has been widely used in the evaluation of many video surveillance anomaly detection works (SULTANI; CHEN; SHAH, 2018; TIAN et al., 2021; LV et al., 2021). We extracted an intermediate layer to be the features of the set of T frames (Figures 19 and 20). The I3D weights in the pretext task were pre-trained on Kinects (CARREIRA; ZISSERMAN, 2017).

Our proposed SSOC method applied in the I3D backbone (SSOC+I3D) was compared with the I3D network pre-trained on the Kinect dataset. The results is found on Section 5.2. Note that the models are the same, the difference is that our SSOC+I3D followed the novel pre-training procedure described in Section 19.

5.2 Results

This Section presents the results of the SSOC method, using the experimental setup described in Section 5.1.3. Section 5.2.1.1 describes the results for the UCF-Crime dataset, and Section 5.2.1.2 describes the results for the CamNuvem dataset.

5.2.1 General Evaluation

Tables 6 and 7 present the AUC values for the UCF-Crime dataset and CamNuvem dataset, respectively. The tables are divided into two parts: (i) All test set, which represents the AUC values calculated over all samples in the test partition, and (ii) Only abnormal videos, which represents the AUC values calculated over only the abnormal samples in the test partition. Using method (ii) allows for a better measurement of how well a method performs at localizing the anomaly in the video. The first column represents the approach used to acquire the feature vectors from the videos, namely I3D and SSOC+I3D. The second, third, and fourth columns present the values using the RTFM, WSAL, and RADS methods, respectively. All values were calculated after 5 runs. In each cell, the highest value is depicted, followed by the mean and standard deviation in parentheses.

5.2.1.1 UCF-Crime dataset

Note that in Table 6, considering the All test set partition, the RTFM method achieved an AUC of 0.747 using the I3D feature vector, while it achieved an AUC of 0.734 using the SSOC+I3D feature vector. This represents a worsening of 0.013 in AUC, which is a 1.3% absolute worsening. The WSAL method achieved an AUC of 0.723 using the I3D feature vector, while it achieved an AUC of 0.751 using the SSOC+I3D feature vector. This represents an improvement of 0.028 in AUC, which is a 2.8% absolute improvement. The RADS method achieved an AUC of 0.700 using the I3D feature vector, while it achieved an AUC of 0.728 using the SSOC+I3D feature vector. This represents an improvement of 0.028 in AUC, which is a 2.8% absolute improvement. Figure 21a shows the ROC curves for RTFM, WSAL, and RADS using all the samples in the UCF-Crime dataset.

In the Only abnormal videos partition of Table 6, the RTFM method achieved an AUC of 0.540 using the I3D feature vector, while it achieved an AUC of 0.528 using the SSOC+I3D feature vector. This represents a worsening of 0.012 in AUC, which is a 1.2% absolute worsening. The WSAL method achieved an AUC of 0.630 using the I3D feature vector, while it achieved an AUC of 0.633 using the SSOC+I3D feature vector. This represents an improvement of 0.003 in AUC, which is a 0.3% absolute improvement. The RADS method achieved an AUC of 0.479 using the I3D feature vector, while it achieved an AUC of 0.549 using the SSOC+I3D feature vector. This represents an improvement of 0.070 in AUC, which is a 7% absolute improvement. Figure 21b shows the ROC curves for RTFM, WSAL, and RADS using only the abnormal samples in the UCF-Crime dataset.

Table 6 – Results 1: Greater values of AUC from 5 runs using UCF-Crime dataset.

	RTFM	**WSAL**	**RADS**
All test set			
I3D	**0.747** **(0.745 ± 0.002)**	0.723 (0.700 ± 0.016)	0.700 (0.698 ± 0.001)
SSOC+I3D (ours)	0.734 (0.723 ± 0.011)	**0.751** **(0.739 ± 0.008)**	**0.728** **(0.721 ± 0.006)**
Only abnormal videos			
I3D	**0.540** **(0.531 ± 0.007)**	0.630 (0.616 ± 0.012)	0.479 (0.477 ± 0.001)
SSOC+I3D (ours)	0.528 (0.511 ± 0.012)	**0.633** **(0.622 ± 0.008)**	**0.549** **(0.539 ± 0.006)**

Table 7 – Results 2: Greater values of AUC from 5 runs using CamNuvem dataset.

	RTFM	**WSAL**	**RADS**
All test set			
I3D	0.858 (0.848 ± 0.006)	0.807 (0.798 ± 0.006)	**0.791** **(0.788 ± 0.002)**
SSOC+I3D (ours)	**0.884** **(0.878 ± 0.003)**	**0.838** **(0.837 ± 0.001)**	0.776 (0.772 ± 0.002)
Only abnormal videos			
I3D	0.560 (0.522 ± 0.021)	**0.628** **(0.598 ± 0.018)**	**0.519** **(0.518 ± 0.001)**
SSOC+I3D (ours)	**0.616** **(0.601 ± 0.014)**	0.573 (0.562 ± 0.009)	0.467 (0.465 ± 0.001)

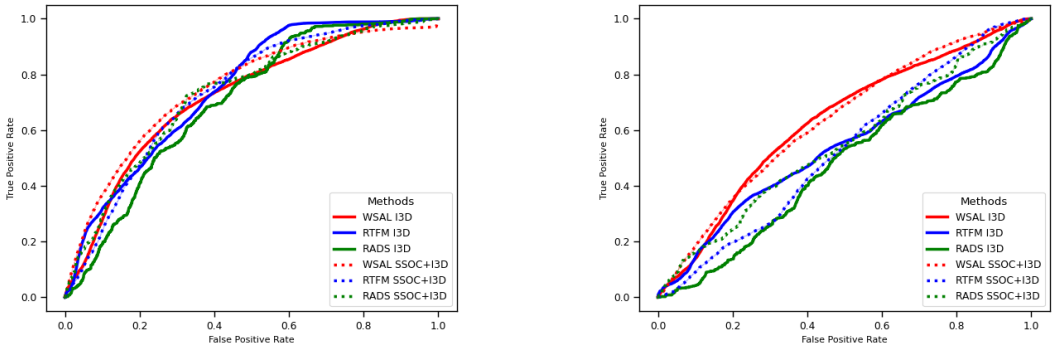
5.2.1.2 CamNuvem dataset

Note that in Table 7, considering the All test set partition, the RTFM method achieved an AUC of 0.858 using the I3D feature vector, while it achieved an AUC of 0.884 using the SSOC+I3D feature vector. This represents an improvement of 0.026 in AUC, which is a 2.6% absolute improvement. The WSAL method achieved an AUC of 0.807 using the I3D feature vector, while it achieved an AUC of 0.838 using the SSOC+I3D feature vector. This represents an improvement of 0.031 in AUC, which is a 3.1% absolute improvement. On the other hand, the RADS method achieved an AUC of 0.791 using the I3D feature vector, while it achieved an AUC of 0.776 using the SSOC+I3D feature vector. This represents a worsening of 0.015 in AUC, which is a 1.5% absolute worsening. Figure 22a shows the ROC curves for RTFM, WSAL, and RADS using all the samples in the test set.

In the Only abnormal videos partition of Table 7, the RTFM method achieved an AUC of 0.560 using the I3D feature vector, while it achieved an AUC of 0.616 using the SSOC+I3D feature vector. This represents an improvement of 0.056 in AUC, which is a 5.6% absolute improvement. The WSAL method achieved an AUC of 0.628 using the I3D feature vector, while it achieved an AUC of 0.573 using the SSOC+I3D feature vector. This represents a worsening of 0.055 in AUC, which is a 5.5% absolute worsening. The RADS method achieved an AUC of 0.519 using the I3D feature vector, while it achieved an AUC of 0.467 using the SSOC+I3D feature vector. This represents a worsening of 0.052 in AUC, which is a 5.2% absolute worsening. Figure 22b shows the ROC curves for RTFM, WSAL, and RADS using only the abnormal samples in the test set.

Figure 21 – Roc curve for I3D and I3D+SSOHC using the methods WSAL, RTFM, and RADS.

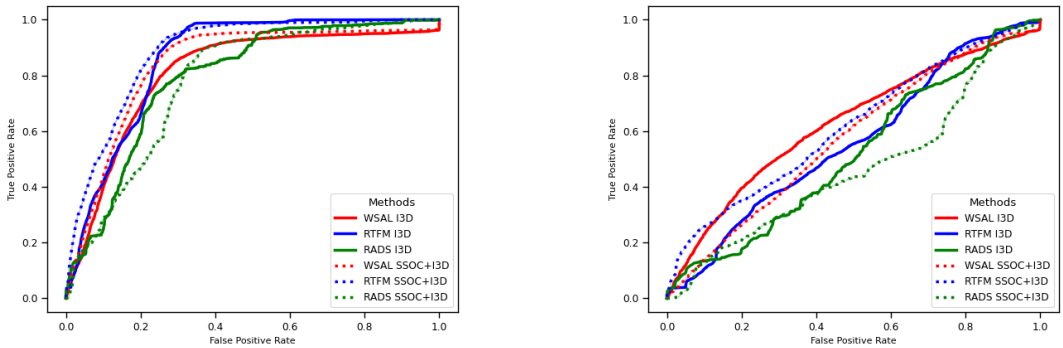
(a) All test set (b) Only abnormal videos



Source: Author's own work.

Figure 22 – Roc curve for I3D and I3D+SSOHC using the methods WSAL, RTFM and RADS.

(a) All test set (b) Only abnormal videos



Source: Author's own work.

5.3 Discussion and Conclusions

We proposed the SSOC approach to generate high-quality feature vectors from videos. We evaluated our approach alongside the traditional I3D feature extractor and found that, out of the 12 cases evaluated, 7 of them (58%) achieved better results. In particular, the WSAL and RTFM methods, applied to the UCF-Crime and CamNuvem datasets respectively, demonstrated superiority over traditional approaches in all evaluated metrics.

A significant limitation of this work is the label generation process, where the success of the training relies on the quality of the Object Tracking module.

6 Self-Supervised Learning for Extracting Interpersonal Dynamics in Video Robbery Detection

This Chapter introduces a Self-Supervised method that leverages the 3D postures of individuals in videos to enhance robbery detection in video surveillance footage. The code is freely available[1]. For our experiments, we employ a vanilla transformer trained in a supervised manner for video robbery detection. To achieve this, we annotated all training videos in the CamNuvem dataset (PAULA; SALVADEO; ARAUJO, 2022). Furthermore, we propose a variation of the CamNuvem dataset and the UCF-Crime dataset, named the CamNuvem Balanced dataset and UCF-Crime Balanced dataset, respectively, to address the issue of imbalanced data.

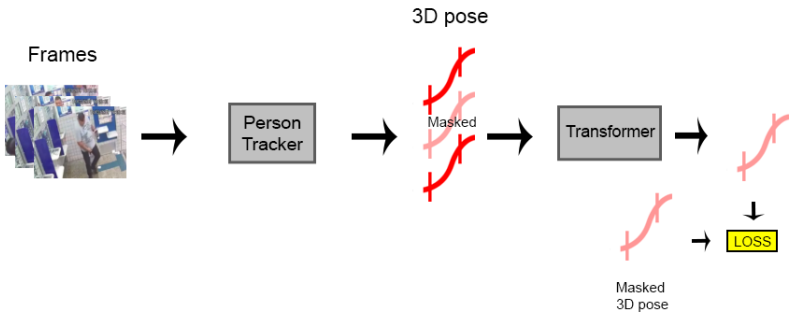
This Chapter is organized as follows: Section 6.1 presents the experimental methodology. Section 6.2 provides the experimental results and discussion. Section 6.3 presents the chapter's conclusions.

6.1 Methodology

Figure 23 depicts an overall definition of the pretext task, while Figure 24 illustrates an overall definition of the downstream task. In both cases, the proposed method begins by re-identifying and extracting the 3D poses of individuals in a video segment using a Person Tracker (the first gray square in the figures). In the pretext task (Figure 23), one person has their 3D pose masked, and the remaining 3D poses are sent to a Transformer, which predicts the 3D pose of the occluded individual. The loss is calculated using the previously masked 3D pose, and the Transformer is trained. Note that no human labor is required, nor is there a need for dense annotation of classes or actions. In the downstream task (Figure 24), all 3D poses are used as input to the Transformer, which estimates whether the scene contains a robbery.

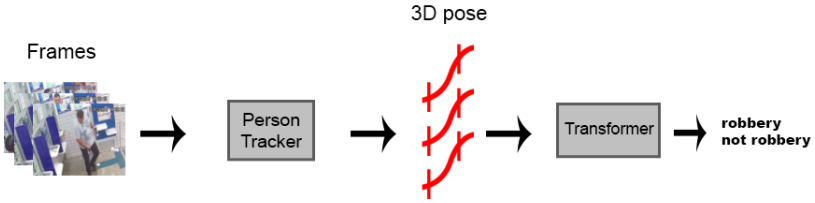
[1] Code, dataset and annotations are freely available at: <https://github.com/daviduarte/Multi-Person-Relationship-Extraction-for-Video-Surveillance-Robbery-Detection.>

Figure 23 – Overview of the pretext task.



Source: Author's own work.

Figure 24 – Overview of the downstream task.



Source: Author's own work.

In the following sections, this procedure will be further developed, and formal definitions will be provided. This section is organized as follows: Section 6.1.1 describes the dataset used. Section 6.1.2 explains the basic definitions needed to understand the proposed method. Section 6.1.3 presents the proposed pretext task. Section 6.1.4 discusses the ability of our method to create new samples in the pretext task. Section 6.1.5 explains how transfer learning is applied to the downstream task.

6.1.1 Dataset

In the experiments, the CamNuvem (PAULA; SALVADEO; ARAUJO, 2022) and UCF-Crime (SULTANI; CHEN; SHAH, 2018) datasets are used to evaluate the results. Sections 6.1.1.1 and 6.1.1.2 describe how they are used in the experiments, respectively.

6.1.1.1 CamNuvem dataset

This dataset consists of real-world robbery videos acquired from video surveillance cameras. It is weakly labeled, meaning each video in the training set has only one label: 1 if there is an abnormal segment somewhere in the video, and 0 otherwise. To enable supervised

training of a robbery detector, we manually labeled all training videos in the CamNuvem dataset, identifying the exact start and end frames for each anomaly.

The videos in the CamNuvem dataset are categorized into anomaly and normal categories, with each category containing 437 videos in the training set and 49 videos in the test set. In the normal category, all video frames are normal, whereas in the anomaly category, not all frames are anomalous due to the presence of normal segments within these videos. This causes an imbalance in the dataset classes because the "normal" class has significantly more frames than the "anomaly" class. Table 8 shows the number of frames for the training partition, and Table 9 shows the number of frames for the test partition of the CamNuvem dataset. It is evident that there are far more normal frames than anomaly frames.

Table 8 – Number of frames in the training partition of the CamNuvem dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	4,917,564	0
Anomaly Videos	440,109	575,055
Total:	**5,357,673**	**575,055**

Table 9 – Number of frames in the test partition of the CamNuvem dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	115,128	0
Anomaly Videos	41,108	63,724
Total:	**156,236**	**63,724**

To mitigate this issue and prevent evaluation problems caused by unbalanced data, we created a new dataset named CamNuvem Balanced, which includes smaller normal videos to equalize the total number of normal and anomaly frames. Table 10 presents the number of frames in this new dataset for the training partition, while Table 11 presents the number of frames for the test partition. The CamNuvem Balanced dataset and the hard anomaly annotations are freely available and will be used in the experiments[2].

[2] The CamNuvem Balanced dataset and the hard anomaly annotations are freely available at: <https://github.com/daviduarte/camnuvem-dataset>.

Table 10 – Number of frames in the training partition of the CamNuvem Balanced dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	135,198	0
Anomaly Videos	439,875	575,055
Total:	**575,073**	**575,055**

Table 11 – Number of frames in the test partition of the CamNuvem Balanced dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	22,644	0
Anomaly Videos	41,090	63,724
Total:	**63,734**	**63,724**

6.1.1.2 UCF-Crime dataset

The UCF-Crime dataset contains real-world videos acquired from surveillance cameras, depicting 13 types of situations: Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Road Accidents, Robbery, Shooting, Shoplifting, Stealing, and Vandalism. The videos are weakly labeled, meaning each sample has one label indicating whether there is an anomaly in the video, but the exact location of the anomaly is not known. This dataset also contains normal videos that depict routine situations without anomalies. However, the work of Maqsood et al. (2021) provides frame-level annotations for the videos in UCF-Crime, which are used in the experiments. Furthermore, in the experiments, these different categories are merged into just one category. In this way, a video segment can be labeled as either “anomaly” or “normal”.

Similar to the CamNuvem Balanced dataset, we created a modified version of this dataset, named UCF-Crime Balanced, to address the problem of data imbalance between the normal and anomalous frames. Due to the human-centric nature of our method, we excluded the video categories “Road Accidents” and “Explosion” because these videos do not depict humans and are not useful for this work. Table 12 shows the number of frames for the training partition, and Table 13 shows the number of frames for the test partition of the UCF-Crime dataset. It is evident that there are far more normal frames than anomaly frames.

Table 14 presents the number of frames in UCF-Crime Balanced for the training partition, while Table 15 presents the number of frames for the test partition. The UCF-Crime Balanced

dataset is freely available and will be used in the experiments[3].

Table 12 – Number of frames in the training partition of the UCF-Crime dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	9,477,864	0
Anomaly Videos	1,960,531	1,222,733
Total:	**11,438,395**	**1,222,733**

Table 13 – Number of frames in the test partition of the UCF-Crime dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	649,152	0
Anomaly Videos	339,342	123,834
Total:	**988,494**	**123,834**

Table 14 – Number of frames in the training partition of the UCF-Crime Balanced dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	935,586	0
Anomaly Videos	281,941	1,222,733
Total:	**1,217,527**	**1,222,733**

Table 15 – Number of frames in the test partition of the UCF-Crime Balanced dataset.

	Normal frames	**Anomaly Frames**
Normal Videos	100,098	0
Anomaly Videos	33,178	114,224
Total:	**133,276**	**114,224**

6.1.2 Definitions

Given a video, consider a temporal window, or scene, denoted as S, containing a set of T consecutive frames, where $S = \{1, 2, \ldots, T\}$. Consider also a person $i \in \{0, 1, \ldots, N\}$ who is re-identified and tracked in the scene using PHALP (Predicting Human Appearance, Location

[3] UCF-Crime Balanced is available at: <https://github.com/daviduarte/camnuvem-dataset>.

and Pose for Tracking) (RAJASEGARAN et al., 2022). This implies that for a person i at frame $t \in S$, it is possible to recover and track that person in any other frame within the scene. Here, N represents the maximum number of people detected in each frame t. If a scene contains more than N people, only the first N are considered, and any additional individuals are discarded. If a scene lacks any persons, the entire scene is disregarded. It is important to note that, in this work, only persons are considered; however, other object categories, such as "cars" and "weapons" may be incorporated using the approach outlined in this paper

Considering H_t^i as the representation (feature vector) of person i in frame t, and given that person i is tracked and re-identified within the scene, one can gather all these representations, resulting in Φ_i, a spatio-temporal (tubelet) representation for person i:

$$\Phi_i = \{H_1^i, H_2^i, ..., H_T^i\}. \tag{6.1}$$

In this work, the person's posture is used as a feature vector. Three approaches are used to represent a person's posture in 3D space: the Skinned Multi-Person Linear Model (SMPL) method (LOPER et al., 2023), the raw 3D joint coordinates of individuals in the scene using the PHALP method, and a concatenation of these two. Further details are provided in the following sections.

6.1.2.1 SMPL

This approach represents each person in 3D space using the SMPL method. For a given person i at frame t, the feature vector H_t^i is defined as:

$$H_t^i = P_t^i, \tag{6.2}$$

where $P_t^i \in \mathbb{R}^{229}$ is generated by the SMPL, representing the 3D pose and shape of the person. It consists of 207 parameters relating to joint angles ($23 \times 3 \times 3$), 9 parameters related to the global orientation of the person, 10 parameters for body shape, and 3 parameters for the 3D translation of the person in the camera frame. When using SMPL as feature vectors, the tubelet for person i throughout the scene is denoted as Φ_i^P:

$$\Phi_i^P = \{H_1^i, H_2^i, ..., H_T^i\}. \tag{6.3}$$

6.1.2.2 Joints 3D

This approach represents each person in 3D space using raw 3D joint coordinates generated by the PHALP method. For a given person i at frame t, the feature vector H_t^i is defined as:

$$H_t^i = J_t^i, \tag{6.4}$$

where $J_t^i \in \mathbb{R}^{135}$ represents the 3D pose of the person, composed of 45 joints (e.g., elbow, knee, hands, etc.) multiplied by their 3D coordinates, resulting in a vector of size 135. The choice of 3D joints over the 2D version is based on evidence of better performance in action recognition (RAJASEGARAN et al., 2021). When using 3D joints as feature vectors, the tubelet for person i throughout the scene is denoted as Φ_i^J:

$$\Phi_i^J = \{H_1^i, H_2^i, ..., H_T^i\}. \tag{6.5}$$

6.1.2.3 SMPL + Joints 3D

This approach represents each person in 3D space using the vector generated by SMPL (Section 6.1.2.1), concatenated with the raw 3D joint coordinates generated by the PHALP method (Section 6.1.2.2). For a given person i at frame t, the feature vector H_t^i is defined as:

$$H_t^i = C_t^i, \tag{6.6}$$

considering $C_t^i \in \mathbb{R}^{364}$ as the concatenation of $P_t^i \in \mathbb{R}^{229}$ with $J_t^i \in \mathbb{R}^{135}$. Thus, the tubelet for person i throughout the scene is referred to as Φ_i^C:

$$\Phi_i^C = \{H_1^i, H_2^i, ..., H_T^i\}. \tag{6.7}$$

6.1.3 Pretext Task

A vanilla Transformer $\mathcal{F}$, following the methodology of LART (Lagrangian Action Recognition with Tracking) (RAJASEGARAN et al., 2023), is used in the experiments. This model is a Transformer (VASWANI et al., 2017) consisting of 16 layers with a width of 512. The architecture includes 16 self-attention heads and is followed by layer normalization (BA; KIROS;

HINTON, 2016). At the top of the architecture, for the pretext task, there is a Fully Connected Network (FCN) with 1 layer, and for the downstream task, there is an FCN with 4 layers.

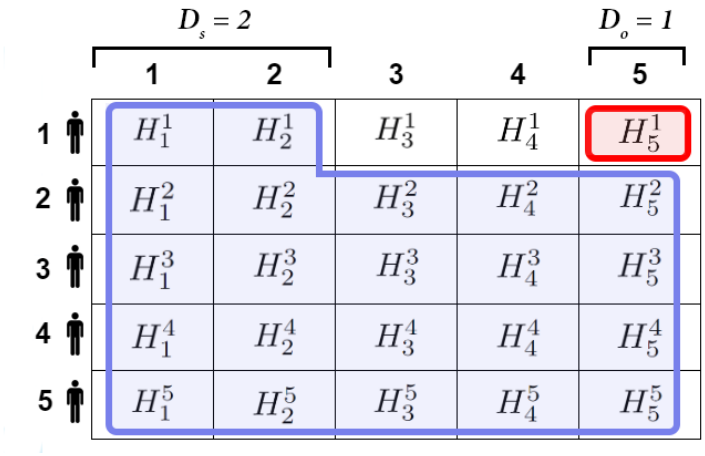
The model $\mathcal{F}$ is employed to predict the future feature vector of a person. This prediction is based on the person's past feature vector and the feature vectors of other individuals in the scene. Masked attention is applied in cases where detection is missing due to issues such as occlusion. The selected person for predicting the future feature vector is called the target person, denoted as $TS \in \{1, 2, ..., N\}$. In this context, a portion of the target person's tubelet, represented as Φ_{TS}, is intentionally occluded and serves as the target for the model $\mathcal{F}$. Another portion of the target person's tubelet is used as part of the input for the model $\mathcal{F}$.

Figure 25a illustrates a hypothetical scene with 5 persons (represented by lines) and 5 frames (represented by columns) in a grid. Each cell in the grid corresponds to the feature vector of a given person for a specific frame. In this figure, person 1 is chosen as the target person. During the pretext task training, certain future feature vectors (indicated by the red cell) are predicted using the person's past feature vectors and the feature vectors of other people in the scene (indicated by the blue cells). In Figure 25b, the pretext task training process of model $\mathcal{F}$ is represented, with the blue cells as the model's input and the red cell as the model's output. This training process uses only the information already available in the dataset, eliminating the need for human labor to label the target (red cell). The sole requirement is a tracking algorithm (in this case, PHALP (RAJASEGARAN et al., 2022) is used) to re-identify individuals between frames. The detected people's tubelets may vary in size, either because they exit the scene before it ends or enter the scene after it begins. For simplicity, only the tubelets with a size equal to T are considered.

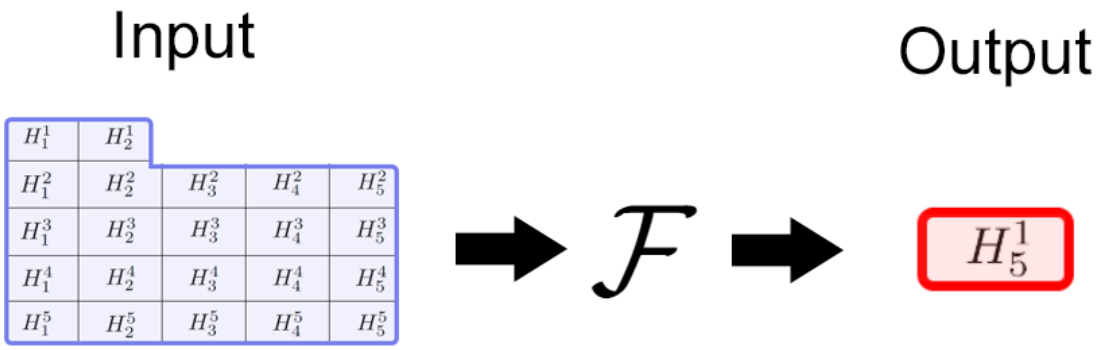
Consider D_s as the size of the first part of the target person TS (blue part) and D_o as the size of the last part of the target person (red part). In Figure 25a, $D_s = 2$ and $D_o = 1$. The first part of the target person is used as part of the input, while the last part is used as the target. Note that there is a gap in the middle of the target person's tubelet, compelling the model to infer the future of this person.

Let so and eo denote the start and end indexes of the target feature vector in Φ_{TS} during pretext task training. Then, $\check{\Phi}$ and $\hat{\Phi}$ are defined as the first part of Φ_{TS} (included as part of the input to the model $\mathcal{F}$) and the last part of Φ_{TS} (used as the target in model training), respectively. They are defined as follows:

Figure 25 – Training procedure. (a) The representation of feature vectors H^i_j for 5 persons (rows) across 5 frames (columns) in a hypothetical scene. Person 1 (first row) is the target person, with some future feature vectors (red cells) utilized as targets. The past feature vector of person 1, along with the feature vectors of other individuals in the scene (blue cells), is used as input. (b) All the feature vectors represented by the blue cells in (a) are concatenated and made available as input for model $\mathcal{F}$, while the feature vectors represented by the red cells are used as targets. It is important to note that no human-generated labels are used in this pretext task.



(a)



(b)

Source: Author's own work.

$$\check{\Phi} \subset \Phi_{TS}, \check{\Phi} = \{\Phi_{TS_k} | k \in \{1, 2, ..., so - 1\}\} \tag{6.8}$$

$$\hat{\Phi} \subset \Phi_{TS}, \hat{\Phi} = \{\Phi_{TS_k} | k \in \{so, so + 1, ..., eo\}\} \tag{6.9}$$

In the pretext task, the goal is to predict the target person's future feature vector, given its own past features and the features of other people in the scene. Thus, the pretext is defined as follows:

$$\mathcal{F}(\check{\Phi}, ..., \Phi_k, ...; \theta) = \hat{\Phi}, \tag{6.10}$$

where $k \in \{1, 2, \ldots, N\}$ and $k \neq TS$, because Φ_k represents the other people in the scene. Furthermore, θ represents the parameters learned during the training process. The idea of this pretext task is that the relationships among all people in the scene are important for determining the visual appearance of some individuals in the future. By structuring the pretext task in this manner, the model can capture the intrinsic social behavior of people.

The pretext task described in Equation 6.10 can be instantiated using the SMPL, Joints 3D, and SMPL + Joints 3D as features, employing the specific notations for these features as described in Sections 6.1.2.1, 6.1.2.2, and 6.1.2.3. This results in the three pretext tasks proposed by this work:

$$\mathcal{F}(\check{\Phi}^P, ..., \Phi_i^P ...; \theta) = \hat{\Phi}^P, \tag{6.11}$$

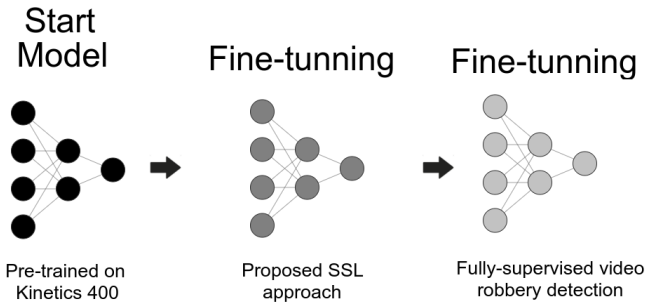
$$\mathcal{F}(\check{\Phi}^J, ..., \Phi_i^J ...; \theta) = \hat{\Phi}^J, \tag{6.12}$$

$$\mathcal{F}(\check{\Phi}^C, ..., \Phi_i^C ...; \theta) = \hat{\Phi}^C, \tag{6.13}$$

where $\check{\Phi}^P$, $\check{\Phi}^J$, and $\check{\Phi}^C$ are the $\check{\Phi}$ (defined in Equation 6.8 and represented by H_1^1 and H_2^1 in Figure 25a) using SMPL, Joints 3D, and SMPL + Joints 3D as features, respectively. Similarly, $\hat{\Phi}^P$, $\hat{\Phi}^J$ and $\hat{\Phi}^C$ are the $\hat{\Phi}$ (defined in Equation 6.9 and represented by H_5^1 in Figure 25a) using SMPL, Joints 3D, and SMPL + Joints 3D as features, respectively.

In the pretext task, as shown by Figure 26, the model is pre-trained on the Kinetics-400 dataset (Figure 26 in "Start Model"). After, all layers are fine-tuned using the methods described in this Section (Figure 26 in the First "Fine-tunning"), and the learning rate is set experimentally to 1×10^{-4}. The dataset used is the CamNuvem Balanced dataset and the UCF-Crime dataset. No test samples are used in pretext training. The videos are divided into scenes of 125 frames (4.17 seconds for videos at 30 fps) without overlap. Consecutive windows do not have space between them. All the experiments were conducted using one GPU GeForce RTX 4090 with 24 GB of memory. The sizes D_s and D_o (see Figure 25) are defined according to the limits of the GPU memory, with $D_s = 120$ and $D_o = 5$ for SMPL, $D_s = 115$ and $D_o = 10$ for Joints 3D, and

Figure 26 – The pipeline for the training process.



Source: Author's own work.

$D_s = 122$ and $D_o = 3$ for SMPL + Joints 3D. Mean Square Error (MSE) was used to train the model.

The LART (RAJASEGARAN et al., 2023) method, on which this work is based, considers an appearance vector as input for the transformer; however, the code to extract the appearance vector is not publicly available. Therefore, we do not consider the appearance vector in any of our experiments to avoid appearance bias, and it is filled with zeros when necessary.

6.1.4 Sample Multiplication

In the pretext task, consider a scene with five persons $N = 5$, and the target person randomly selected to be person $TS = 1$. Instantiating Equation 6.10 in this case results in:

$$\mathcal{F}(\check{\Phi}_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5; \theta) = \hat{\Phi}_1. \tag{6.14}$$

The idea here is to predict the future feature vector of the target person given its past feature vector along with the feature vectors of other people in the scene. However, considering the same scene, if the target person TS is selected as a different person, for instance, $TS = 2$, the result is a different instance of Equation 6.10:

$$\mathcal{F}(\check{\Phi}_2, \Phi_1, \Phi_3, \Phi_4, \Phi_5; \theta) = \hat{\Phi}_2, \tag{6.15}$$

where the idea is similar, to predict the future feature vector of target person 2 given its past feature vector along with the feature vectors of other people in the scene. Following this procedure, it is possible to predict the future feature vectors of the different N detected persons in the scene. In

this way, a single scene can generate N samples without human labor. This procedure is applied in the pretext training.

One might question what the optimal value of N is for the pretext task., i.e., how many persons to consider when predicting the future of the target person. This evaluation is conducted in Section 6.2.3, where the value of N is varied in the pretext task, and the impact on the downstream task is measured. In the downstream task, N is fixed at $N = 5$.

6.1.5 Downstream Task

After the model $\mathcal{F}$ is trained in the pretext tasks defined in Section 6.1.3, it is time to fine-tune it in the downstream task, which is supervised robbery detection using the CamNuvem Balanced dataset and supervised anomaly detection using UCF-Crime Balanced dataset (Section 6.1.1), represented by the second "Fine-tunning" in the Figure 26. For this, the classification head is removed and replaced by a downstream head. Now, the expected output is only one value $\hat{Y} \in \{0, 1\}$, denoting whether a scene contains a robbery/anomaly (1) or not (0), and the input is the SMPL concatenated with the joints 3D. The equation for the downstream task is given by:

$$\mathcal{F}'(\Phi_1^C, \Phi_2^C, \Phi_3^C, ..., \Phi_N^C, \theta) = \hat{Y}, \tag{6.16}$$

where $\mathcal{F}'$ is the model $\mathcal{F}$ with the modified classification head.

The idea here is that the model observes how people interact with each other in the scene to assign a robbery/anomaly score. In this task, the Binary Cross Entropy Loss is used. The Area Under Curve (AUC) metric is employed to evaluate the experiments. If the pretext task is beneficial for the model, it will be reflected in good AUC values in this downstream task. Section 6.2 will present the results. In the downstream task, all the model's layers are fine-tuned, and the learning rate is set to 0.7×10^{-4}. No samples from the test partition are used to train the model. All the AUC values are the best ones found in 5 runs, followed by the mean and the standard deviation in parentheses.

In Section 6.1.4, one could note that the proposed pretext tasks can multiply the training set by creating new samples. In this way, if a small dataset has few unlabeled samples as well as labeled samples, the proposed pretext tasks could improve model performance due to this characteristic of creating more samples. To check this hypothesis, this work separates the experiments into two parts: the first, described in Section 6.1.5.1, divides the dataset into splits

containing portions of labeled and unlabeled samples without overlapping. The second, described in Section 6.1.5.2, performs pretext task training using the same training set as the downstream task training, but with the labels ignored.

6.1.5.1 Training On Splits

To better understand the benefit of the proposed pretext task described in Section 6.1.3, the dataset used in the experiments was divided into three splits: 10%, 50%, and 80%. Each split contains a proportion of labeled samples in the training set, and the labels of the remaining samples are ignored. For example, the 10% split contains 10% labeled samples, while the remaining 90% are kept unlabeled. The pretext task is trained using the 90% of unlabeled samples, and the downstream task is conducted using the 10% of labeled samples. The labeled and unlabeled parts are disjoint. The same approach is applied to the other splits, with the selection of samples kept in the labeled part being random. The test set remains unchanged and is the same for all splits.

6.1.5.2 Training On The Same Set

In this experiment, there are no splits. The pretext task is trained using the entire training set (100% of samples) with the labels ignored. Subsequently, in the downstream task, fine-tuning is performed using the same samples, but this time utilizing the labels.

6.2 Results and Discussions

This section is organized as follows: Section 6.2.1 discusses the results for the Training on Splits methodology. Section 6.2.2 covers the results for the Training on the Same Set methodology. Section 6.2.3 examines the influence of the number of people in the pretext task. Section 6.2.4 addresses the number of samples generated by the proposed pretext tasks for each N value.

6.2.1 Results for Training on Splits

In this section, the results of the experiments explained in Section 6.1.5.1 are discussed. Table 16 presents the AUC results for the downstream task using the CamNuvem Balanced dataset (values computed using the test set). The AUC values are computed for multiple approaches (rows). The second column, “Start Model”, shows where this approach was pre-trained (as shown

in Figure 26 under "Start Model"). The third column, "Pre-train data", shows where the approach was fine-tuned (as shown in Figure 26 under the first "Fine-tuning"). The columns "0%", "50%", and "80%" indicate the dataset split on which the approach was trained.

The second row, "Transformer", shows the baseline, in which the model $\mathcal{F}$ is pre-trained on Kinetics-400 but not fine-tuned using our method (without the first "Fine-tuning" in Figure 26). The third row displays the results for the model pre-trained using the pretext task "3D Joints Only". The fourth row presents the results for the model pre-trained using the pretext task "SMPL Only". The fifth row illustrates the results for the model pre-trained using the pretext task "3D Joints + SMPL".

Table 16 – AUC Results for Multiple Pretext Tasks Across Different Dataset Splits. All methods were pre-trained on Kinetics-400. CNB is CamNuvem Balanced dataset.

	Start Model	**Pre-train data**	**10%**	**50%**	**80%**
Transformer (baseline)	Kinetics-400	-	69.41 (67.96 ± 0.1)	68.58 (67.35 ± 0.01)	70.73 (70.01 ± 0.004)
3D Joints Only	Kinetics-400	CNB	66.66 (65.19 ± 0.01)	70.54 (69.14 ± 0.007)	72.37 (69.60 ± 0.016)
SMPL Only	Kinetics-400	CNB	71.14 (70.25 ± 0.005)	72.75 (71.57 ± 0.01)	68.73 (68.03 ± 0.005)
3D Joints + SMPL	Kinetics-400	CNB	73.42 (71.02 ± 0.03)	71.33 (70.54 ± 0.006)	69.42 (68.60 ± 0.004)

In Table 16, considering the downstream training using only 10% of the data (forth column), it can be noted that our best approach, SMPL + 3D Joints (fifth row), showed an improvement of 4.01 in AUC compared to when the downstream task did not use our approach (Transformer baseline, in first row). This suggests that our approach is useful for extracting knowledge from unlabeled data to improve downstream results.

When considering the downstream task training with only 50% of the data (fifth column), our best approach, SMPL Only (fourth row), achieved an improvement of 4.17 in AUC compared to when the downstream task did not use our approach. In the case where the downstream training used only 80% of the data (sixth column), the 3D Joints Only approach (third row) achieved an improvement of 1.64 in AUC.

Even though some variations of our method presented a worsening in some cases compared with the baseline, the difference is small, and the Transformer baseline was created by pre-training on Kinetics-400, which has hard labels and a large quantity of videos (400–1150

videos per class, with 400 classes). Our method needs far fewer videos (only 972 videos from the CamNuvem Balanced dataset) and no labels to achieve similar or better results.

6.2.2 Results for Training On The Same Set

In this section, the results for the experimental setup explained in Section 6.1.5.2 are discussed. Table 17 summarizes the results using the CamNuvem Balanced dataset as the training set for the downstream task. For comparison, we chose three methods of anomaly detection using Multiple Instance Learning: Real-World Anomaly Detection in Surveillance (RADS) (SULTANI; CHEN; SHAH, 2018) (line two), Robust Temporal Feature Magnitude Learning (RTFM) (LV et al., 2021) (line three), and Weakly Supervised Anomaly Localization (WSAL) (TIAN et al., 2021) (line four), all using I3D (WANG et al., 2018) as feature vectors. Additionally, we fine-tuned a Transformer pre-trained on Kinetics-400 using raw video pixels as input (line five).

Furthermore, the method described in Equation 6.16, pretrained on the Kinetics-400 dataset without the pretext task proposed by this work (without the first "Fine-tuning" in Figure 26), is depicted on line six, which is our baseline. Line seven shows the model trained on the Kinetics-400 dataset and fine-tuned using Pretext Task 1 - SMPL. Line eight shows the model trained on the Kinetics-400 dataset and fine-tuned using Pretext Task 2 - Joints 3D. Line nine shows the model trained on the Kinetics-400 dataset and fine-tuned using Pretext Task 3 - SMPL + 3D Joints.

We also check whether initial pre-training on Kinetics-400 is mandatory for good results. In this regard, line ten of Table 17 depicts the model's "Start Model" initialized with random weights, i.e., skipping the "Start Model" shown in Figure 26. However, this row does use our Pretext Task 3 – SMPL + 3D Joints.

Line eleven depicts the same model, but now fine-tunned using the union of the CamNuvem Balanced and UCF-Crime datasets (not the balanced version). This is done because the balanced versions were created by disregarding some normal samples, which could decrease the final quality of the classifier. In this way, the training can utilize all available samples. Note that the CamNuvem Balanced version is used because the normal videos are the same as those present in the UCF-Crime dataset, so there is no overlap of samples between the two datasets.

The second column shows the type of feature vector used to represent the videos. The third and fourth columns show the model's starting weights and the pre-train data used in the

pretext task ("Start Model" and "Fine-tuning" in Figure 26, respectively). The fifth column shows the Area Under the Curve (AUC) score resulting from the test set of the CamNuvem Balanced dataset. The value depicted is the best AUC found in 5 runs, followed by the mean and the standard deviation in parentheses. The sixth column shows the test speed for each method evaluated.

Table 17 – CamNuvem Balanced AUC Value for the Test Set. *Inference time for blocks of 16 frames. CNB is CamNuvem Balanced dataset and UCF is UCF-Crime dataset.

	Feature	**Start Model**	**Pre-train data**	**AUC**	**Inference speed**
RADS	I3D	Random	-	67.32	13685 segm/sec (2.3 + 0.003) ms
RTFM	I3D	Random	-	69.67	6811 fps (2.3 + 0.049) ms *
WSAL	I3D	Random	-	70.14	6915 fps (2.3 + 0.0139) ms *
Transformer	Raw pixels	Kinetics-400	-	59.04	2564 fps
Transformer (baseline)	Human poses	Kinetics-400	-	74.27 (73.13 ± 0.01)	2.77 fps (360 + 0.964) ms
SMPL only (ours)	Human poses	Kinetics-400	CNB	73.90 (73.10 ± 0.01)	2.77 fps (360 + 0.964) ms
3D joints only (ours)	Human poses	Kinetics-400	CNB	75.21 (73.50 ± 0.01)	2.77 fps (360 + 0.964) ms
3D Joints + SMPL (ours)	Human poses	Kinetics-400	CNB	75.32 (74.96 ± 0.004)	2.77 fps (360 + 0.964) ms
3D Joints + SMPL (ours)	Human poses	Random	CNB	**75.81 (74.61 ± 0.01)**	2.77 fps (360 + 0.964) ms
3D Joints + SMPL (ours)	Human poses	Kinetics-400	CNB + UCF	75.30 (73.73 ± 0.01)	2.77 fps (360 + 0.964) ms

Table 18 shows the results for the UCF-Crime Balanced dataset. Our baseline is a Transformer pre-trained on Kinetics-400 but not fine-tuned using our method. The evaluated method for comparison is our Pretext Task 3, 3D Joints + SMPL, as it achieved the best result on the CamNuvem Balanced dataset. The column "AUC" shows the best AUC value from 5 runs, using the test partition of the UCF-Crime Balanced dataset. The values in parentheses represent the mean and standard deviation, respectively.

Table 18 – UCF-Crime Balanced AUC Value for the Test Set. UCF is UCF-Crime dataset.

	Features	**Start Model**	**Pre-train data**	**AUC**	**Inference speed**
Transformer (baseline)	Human poses	Kinetics-400	-	69.18 (66.97 ± 0.01)	2.77 fps (360 + 0.964) ms
3D Joints + SMPL (ours)	Human poses	Kinetics-400	UCF	65.35 (64.74 ± 0.005)	2.77 fps (360 + 0.964) ms

In Table 17, it can be seen that the proposed approach improved the AUC score for robbery detection. Using the pretext tasks "3D Joints Only" and "3D Joints + SMPL" (with a random start point and pre-trained on the CamNuvem Balanced dataset), the AUC improvements compared to the baseline (Kinetics-400, line 6) were 0.94 and 1.54, respectively. The good performance achieved with a random start point indicates that our method does not need to be pre-trained on Kinetics-400, but only on the CamNuvem Balanced dataset, thus saving computational resources. Note that training on Kinetics-400 uses between 400 and 1,150 labeled videos per class, across 400 classes, which is significantly more videos than the CamNuvem dataset (with a total of 972 videos) used in our approach. In this way, our method stands out for achieving better results using much less data and no human-labeled annotations.

However, in line 11 of Table 17, it can be observed that including UCF-Crime samples in pretext training did not lead to an improvement in AUC. This result may be related to the different domain of the UCF-Crime comparing to CamNuvem dataset.

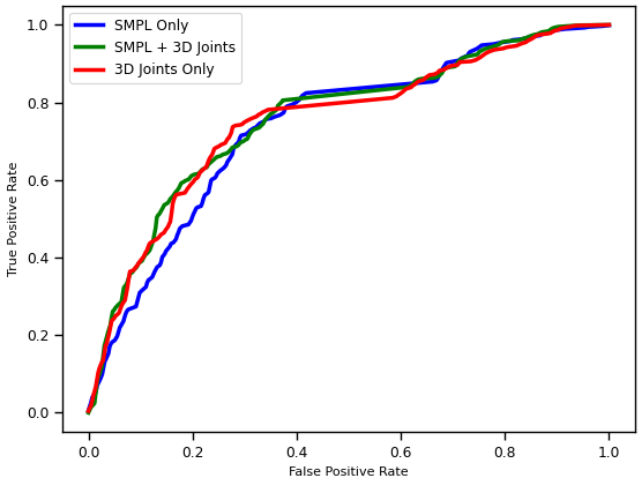
When comparing the results of the "3D Joints + SMPL" method with the best-performing Multiple Instance Learning method (WSAL in line 4 of Table 17), we observe an AUC improvement of 5.67. Note that the WSAL, RTFM, and RADS methods employ specific cost functions designed for video surveillance anomaly detection, whereas our proposed approach uses only a basic cost function for both the pretext and downstream tasks. Therefore, further optimizations to our method could lead to additional improvements in AUC.

Figure 27 shows the ROC curves for all three pretext tasks proposed in this work, and Figure 28 shows the ROC curves for the results in Table 17.

Figures 29 and 30 show a video and the corresponding anomaly score for each scene, using the model pretrained with the "3D Joints + SMPL" pretext task. In both figures, a peak in the anomaly probability can be seen when the person in the video raises their arms. In Figure 29, there are false positive segments, which may correspond to moments when the person places

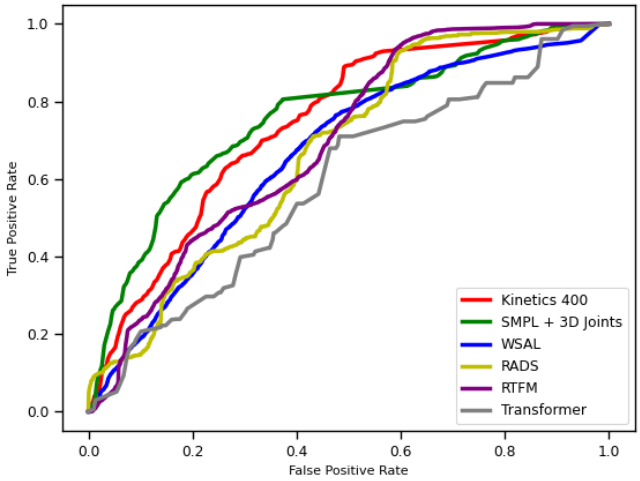
their hands on their waist. In Figure 30, the anomaly probability rises when a weapon appears in the scene.

Figure 27 – ROC Curves for proposed pretext tasks.



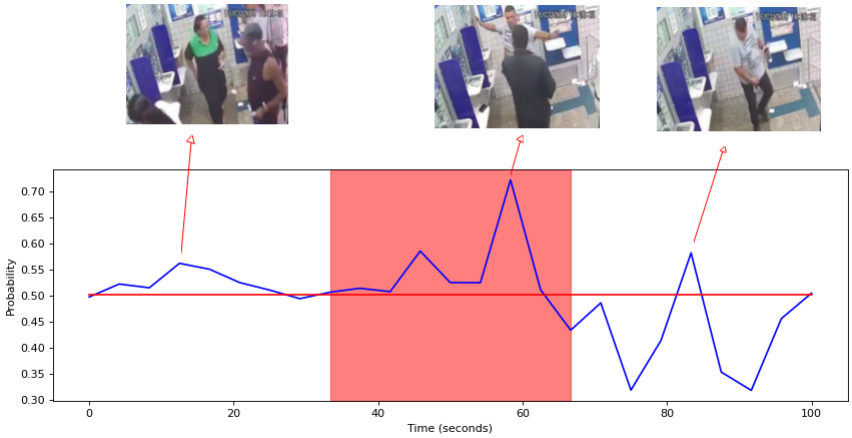
Source: Author's own work.

Figure 28 – ROC Curves for several robbery detection approaches.



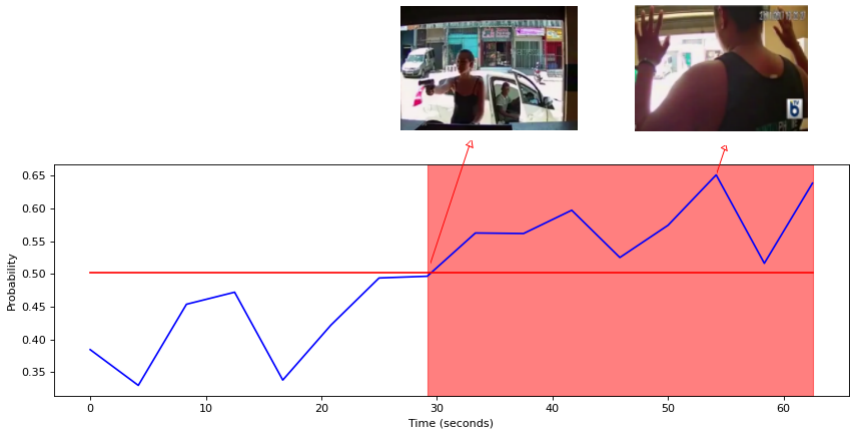
Source: Author's own work.

Figure 29 – Anomaly score (blue line) for the anomaly video 254 from the test set of the CamNuvem Balanced dataset. The classification threshold (red horizontal line) is set to 0.502. The red background indicates the ground truth anomaly segment. The pretext task used was 3D Joints + SMPL.



Source: Author's own work.

Figure 30 – Anomaly score (blue line) for the anomaly video 342 from the test set of the CamNuvem Balanced dataset. The classification threshold (red horizontal line) is set to 0.502. The red background indicates the ground truth anomaly segment. The pretext task used was 3D Joints + SMPL.



Source: Author's own work.

Taking into account the UCF-Crime dataset depicted in Table 18, our method resulted in a decrease of 3.83 in AUC compared to Transformer with start model on Kinetics (the baseline). A possible cause for this decrease in AUC is the complexity of the UCF-Crime dataset, which contains multiple types of anomalies, unlike the CamNuvem dataset, which contains only one type (robbery).

In Table 16, note that 3D Joints Only underperforms compared to SMPL Only in splits

10% and 50%, but outperforms on splits 80% and in Table 17. In a similar way, in Table 16, SMPL + 3D Joints outperforms 3D Joints Only on splits 10% and 50%, underperforms on split 80% and back to outperforms on Table 17. In short, Table 17 depicts a break in the trend presented by Table 16. This occur because in Table 16, for each split of 10%, 50%, and 80%, the samples used in pretext training are not the same as those used in downstream training. On the other hand, in Table 17, the samples used for both the pretext and downstream tasks are the same, specifically 100% of the training set. Therefore, the experimental configuration is different for both tables; in addition, Table 17 uses more samples than Table 16 for both the pretext and downstream tasks. This could affect the trend observed in Table 16. We conducted these experiments to observe whether the re-utilization of samples in pretext and downstream tasks influences the results. As the values shown in Table 17 are generally better than those in Table 16, one can conclude that it is worthwhile to use all labels in pretext task and reuse them in downstream task.

In Table 17, the column "Inference speed" shows the frame inference speed of multiple methods. In the first method, RADS, the speed is depicted in segments per second (segm/sec), unlike frames per second (fps), because the method implementation divided the videos into 32 segments of equal size, independent of video size (SULTANI; CHEN; SHAH, 2018), making the actual fps not directly accessible. However, the speed of the remaining methods is depicted in fps. For all methods, the values depicted in parentheses are the speed of feature extraction and model inference, respectively. In the methods RTFM and WSAL (marked with a *), the speed in parentheses is calculated for segments of 16 frames. For example, for RTFM, it takes 2.3 ms to extract features for 16 frames and 0.049 ms for model inference. In this way, the RTFM fps is calculated as $fps = 1/(((2.3 + 0.049)/16)/1000) = 6811$.

One can note the lower fps of our method (2.77) compared to the other approaches (6811 for RTFM and 6915 for WSAL). This occurred because the human pose extractor used in our work, PHALP (RAJASEGARAN et al., 2022), is not designed to be efficient, but it was chosen due to its great performance on the AVA 2.2 dataset, making it suitable for the proof of concept presented in this work. Note that the model inference time of our approach remains small compared to other methods (0.964 ms compared to 0.049 ms for RTFM and 0.0139 ms for WSAL), and a more efficient human pose extractor could be explored in the future.

6.2.3 People Relationship Study

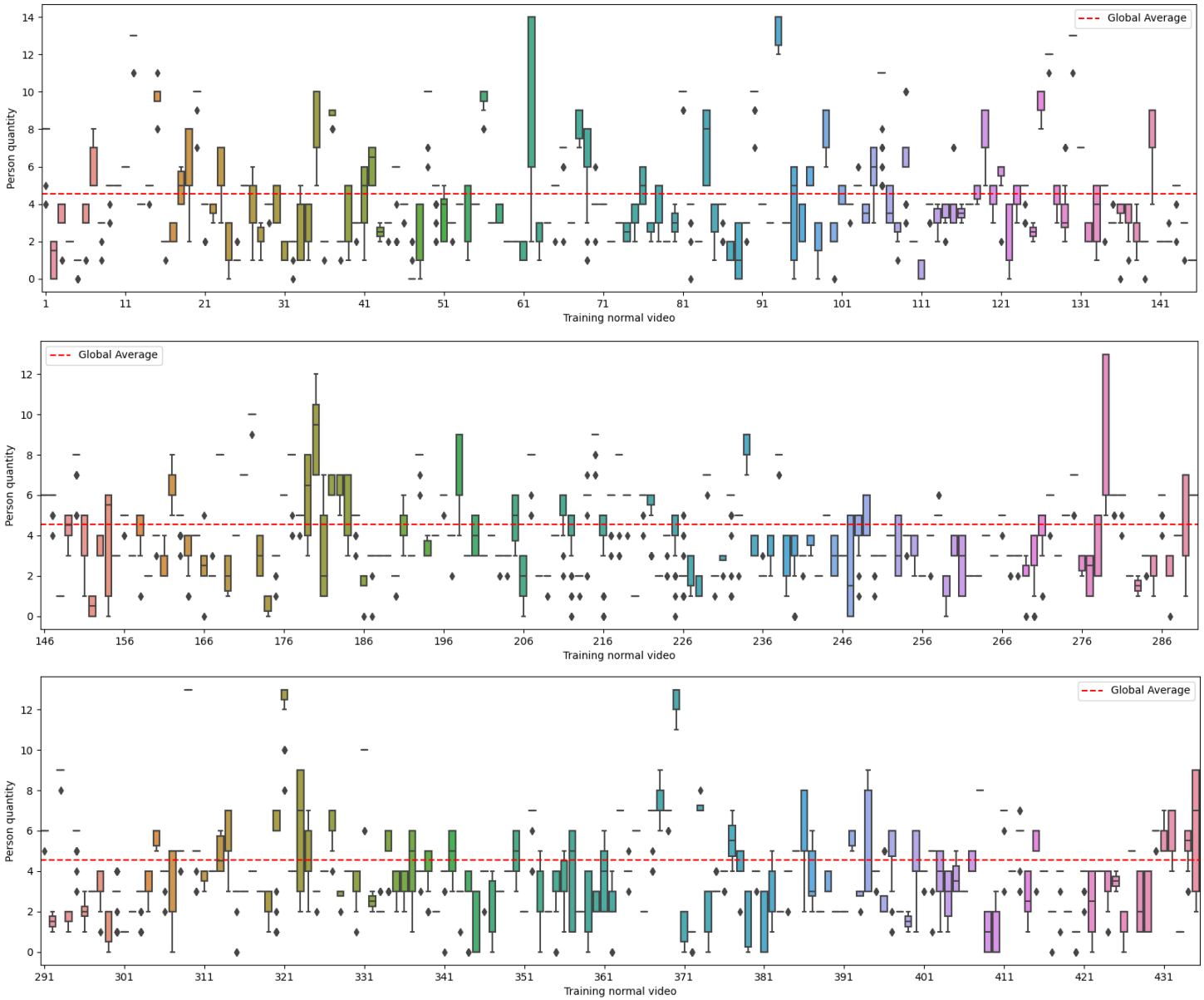
In this section, the choice of the parameter N (Section 6.1.3) is varied to assess its impact on the downstream task. The Pretext Task 3D Joints + SMPL is selected as the base for the experiments. The methodology was defined in Section 6.1.4, i.e., the N value is varied in the pretext task but is fixed at $N = 5$ in the downstream task. In this way, one can verify the real impact of the proposed methodology on the robbery detection task. In Table 19, the first column depicts the N value of the pretext task. The second column shows the test loss acquired by the pretext task. The epoch that achieved this value is used to initialize the weights for the downstream training. The third column depicts the AUC in the downstream task, i.e., video robbery detection using the CamNuvem Balanced dataset.

Table 19 – Individual objects in the input. The results for N=5 are the same as those seen in Table 17.

	Pretext Task Loss	**AUC**
N=1	0.011033	70.40 (70.10 ± 0.002)
N=2	0.003279	72.49 (71.66 ± 0.004)
N=3	0.006178	71.92 (70.96 ± 0.009)
N=4	0.0005584	73.61 (73.12 ± 0.005)
N=5	0.0005584	**75.32 (74.32 ± 0.004)**

From Table 19, it can be noted that the AUC improves as the value of N increases, with the exception of $N = 3$, where there is a break in this trend. This could provide evidence that our method learns the relationships within the scene, and that more people in a scene provide more data to determine whether it is anomalous or normal. To support this hypothesis, the average detected person quantity Q over all segments of all anomaly training videos from the CamNuvem Balanced dataset was calculated, and the obtained value was $Q = 4.56$, which is very close to $N = 5$, which achieved the best AUC value. Figure 31 shows the detected person quantity for each anomaly training video, and the value Q is represented by the horizontal red line. Figures 39, 40, and 41 show the detected person quantity for the normal training videos, test anomaly videos, and test normal videos, respectively.

Figure 31 – The quantity of detected persons (Y-axis) among all 437 anomaly training videos (X-axis). The horizontal red line specifies the number 4.56, which is the average number of detected persons across all scenes in all videos.

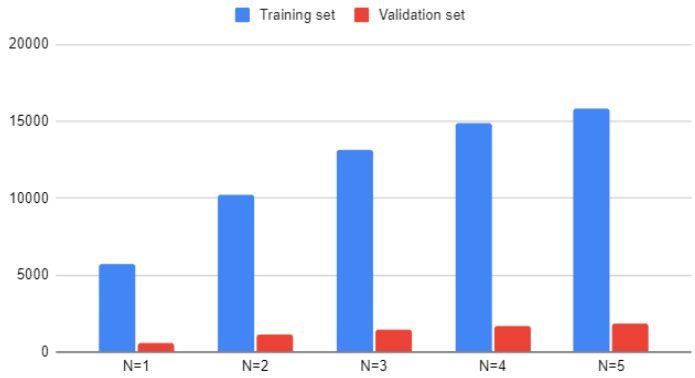


Source: Author's own work.

6.2.4 New data generation analysis

As described in Section 6.1.4, the proposed pretext task multiplies the available number of samples. However, scenes in a video that have no detected persons are ignored. Furthermore, there may be scenes with fewer than N persons detected, which can lead to a decrease in the training sample quantity. Thus, the number of samples available for training for each value of N is shown in Figure 32.

Figure 32 – Sample quantity for all values of N.



Source: Author's own work.

6.3 Chapter Conclusion

Our proposed pretext tasks were designed to understand social behavior in the robbery detection task, aiming to increase the model's performance. The results in Section 6.2.3 show that this is indeed the case; considering N=5 persons in the scene improves results compared to fewer persons, indicating that our model captured the relationships between people on screen.

In our experiments, the downstream task used fully supervised robbery detection. For this purpose, we annotated the CamNuvem dataset and made the hard annotations available. Additionally, we introduced the CamNuvem Balanced dataset and the UCF-Crime Balanced dataset to address the class imbalance problem present in the original datasets, setting a new benchmark for future research in this area.

7 Validating the effectiveness of interpersonal dynamics extraction in the pretext task

In Chapter 6, the reader was introduced to a self-supervised approach to extract interpersonal dynamics from individuals in video surveillance scenes, represented using information about posture. The proposed pretext task was designed to predict the future of a person given all the persons in the past. The idea is that the deep learning model could extract the interpersonal dynamics from the people in the scene to reconstruct the future position of a target person. We observed that this approach improved the downstream task of video robbery detection. However, it is still unknown whether the interpersonal relationships are actually extracted and whether they are useful for reconstructing the target person.

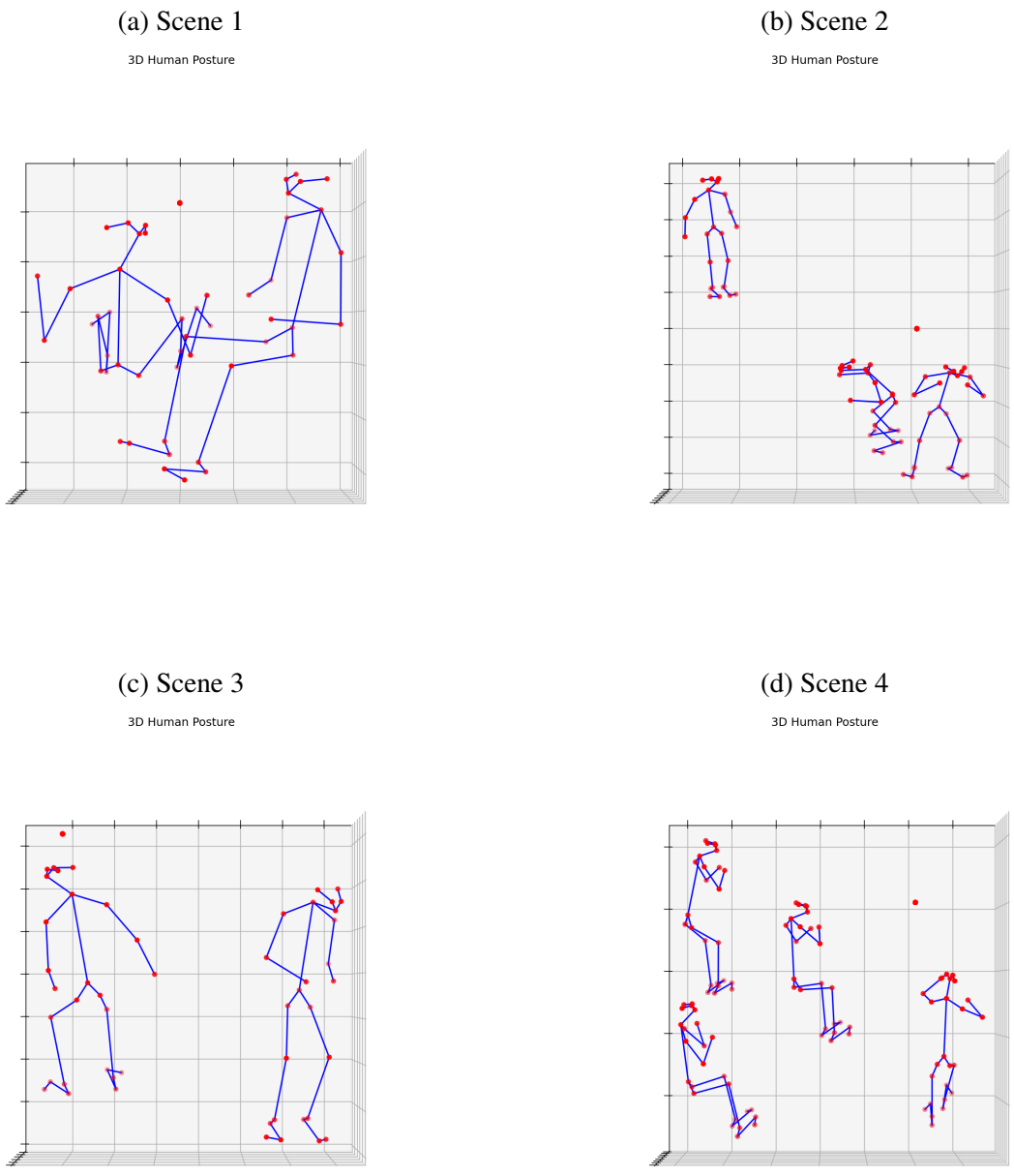
To validate this, we designed the following experiment. In the pretext task, for a given scene and a given target person, we printed both the predicted posture information of the target person (i.e., the output of the deep learning model) and the true value (i.e., the label of the sample). Then, for the same scene and target person, we gradually occluded the other individuals in the scene and observed the reconstruction results. If the method truly extracts interpersonal dynamics, occluding other people should decrease the reconstruction quality of the target person. Conversely, if all individuals other than the target person are occluded and the model does not show a performance drop, then it suggests that other people are not important and the model fails to extract interpersonal dynamics.

For this experiment, the model was trained using a feature vector that combines SMPL and 3D Joints. This means that the feature vector for each person is formed by concatenating SMPL pose information and 3D joint coordinates. However, for this specific analysis, only the 3D joints were considered. Unlike the experiments in previous chapters, in this chapter the dataset was downsampled to 5 fps, while the context size was kept at 125 frames, resulting in a temporal window of $5/125 = 25$ seconds.

The D_o value, depicted in Figure 25a, is set to $D_o = 30$, meaning that the pretext model predicts 30 positions of the target person into the future. However, scenes shorter than the

context size had their last frame repeated until they reached the required length. The repeated frames are not predicted by the pretext task. Therefore, some samples may have fewer than D_o predicted frames. This chapter uses the CamNuvem dataset. Figure 33 illustrates some test samples, showing the lateral projection of the 3D joints of the people in the scenes.

Figure 33 – Test samples from the CamNuvem dataset showing the 3D joints of all detected persons across multiple scenes.



Source: Author's own work.

As explained in Chapter 6, in the pretext task, one person in the scene is selected as the target person, and their future feature vectors are predicted using their past feature vectors along

with the past feature vectors of all other people in the scene. Figure 34 shows some predictions of the pretext task for the scene depicted in Figure 33a, which contains only two individuals. The right-most person was chosen as the target person.

The first row of Figure 34 shows the first, middle, and last predicted 3D poses for the target person, taking into account the presence of the other person in the scene. The second row shows the same predicted poses (first, middle, and last), but these predictions were made disregarding the other individual who shares the scene with the target person—that is, the predictions were made considering only the past of the target person. The third row presents the ground truth for these predictions.

In this case, one can observe that the quality of the reconstruction is very similar when considering only the target person (second row of Figure 34) and when considering both people in the scene (first row of Figure 34).

To represent this numerically, we computed the Mean Squared Error (MSE) between the prediction using the entire scene (first row of Figure 34) and the ground truth, as well as the MSE between the prediction using only the target person and the ground truth, resulting in two values:

$$error_1 = (0.01875, 0.03055, 0.00783) \tag{7.1}$$

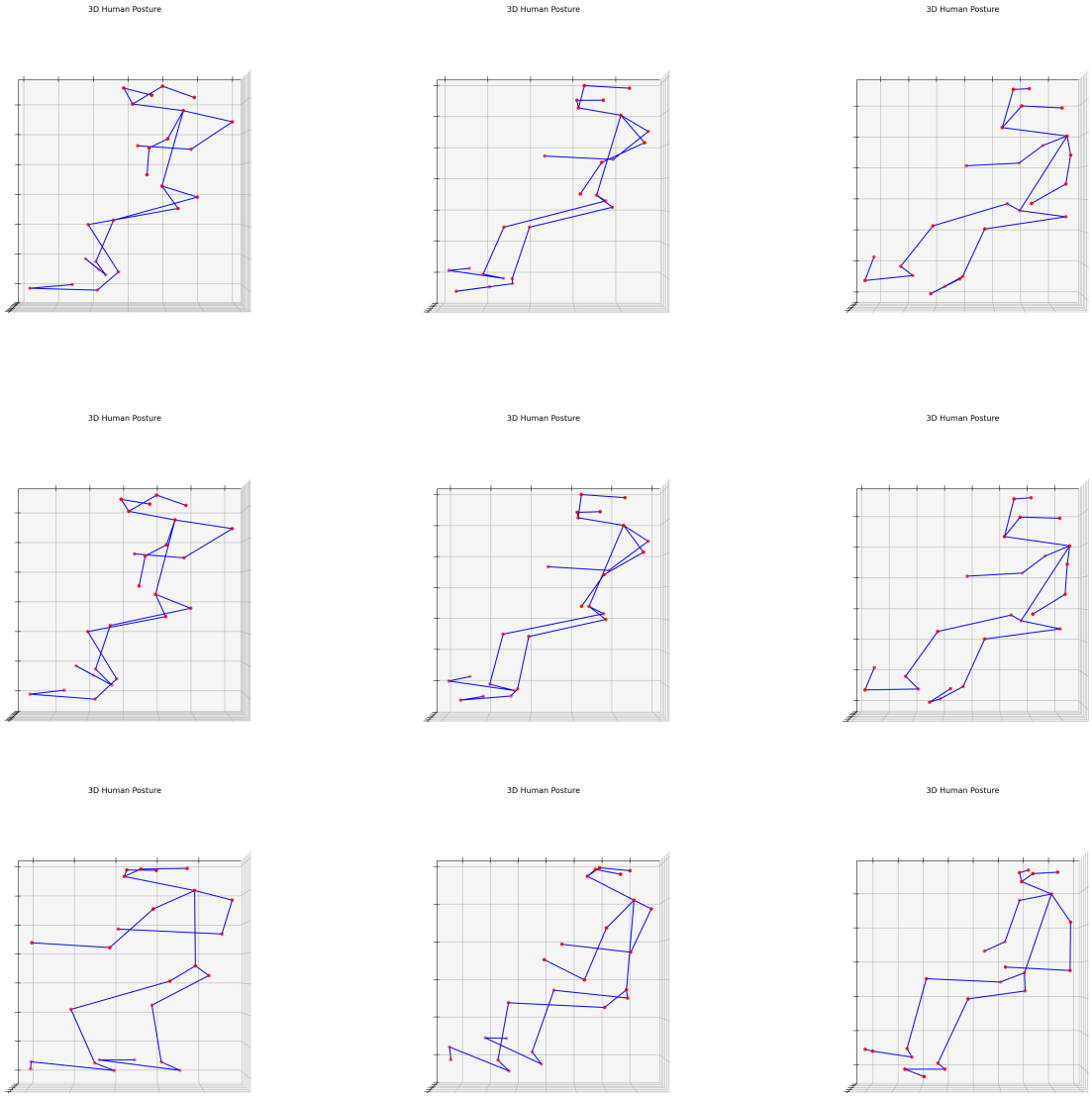
$$error_2 = (0.01766, 0.02959, 0.00887) \tag{7.2}$$

In this context, each of the three components of $error_1$ and $error_2$ represents the MSE for the first, middle, and last reconstructed 3D joints.

Analyzing Figure 34, the reader can observe the similarity between the first and second rows, indicating that including the other person in the scene did not improve the 3D joint reconstruction. Looking at $error_1$ and $error_2$, in Equations 7.1 and 7.2, we see that $error_2$ is smaller than $error_1$ for the first two components. This indicates that including the other person in the scene actually worsened the reconstruction. This suggests that, in this case, no interpersonal dynamics were extracted by the model.

The error was computed by occluding one to four people in the scene, and the mean of these values was calculated. All samples in the test set were considered. Finally, the average of the errors across all scenes was computed, resulting in the following outcome:

Figure 34 – Scene of Figure 33a. Line 1: Reconstruction of the target person considering all persons in the scene. Line 2: Reconstruction of the target person disregarding all other persons in the scene. Line 3: Ground truth for the target person.



Source: Author's own work.

$$values = \begin{bmatrix} 0.01190630 & 0.01542592 & 0.01571143 \\ 0.01257769 & 0.01892536 & 0.01859393 \\ 0.01320453 & 0.01866316 & 0.01859561 \\ 0.01335443 & 0.01872790 & 0.01873018 \end{bmatrix} \tag{7.3}$$

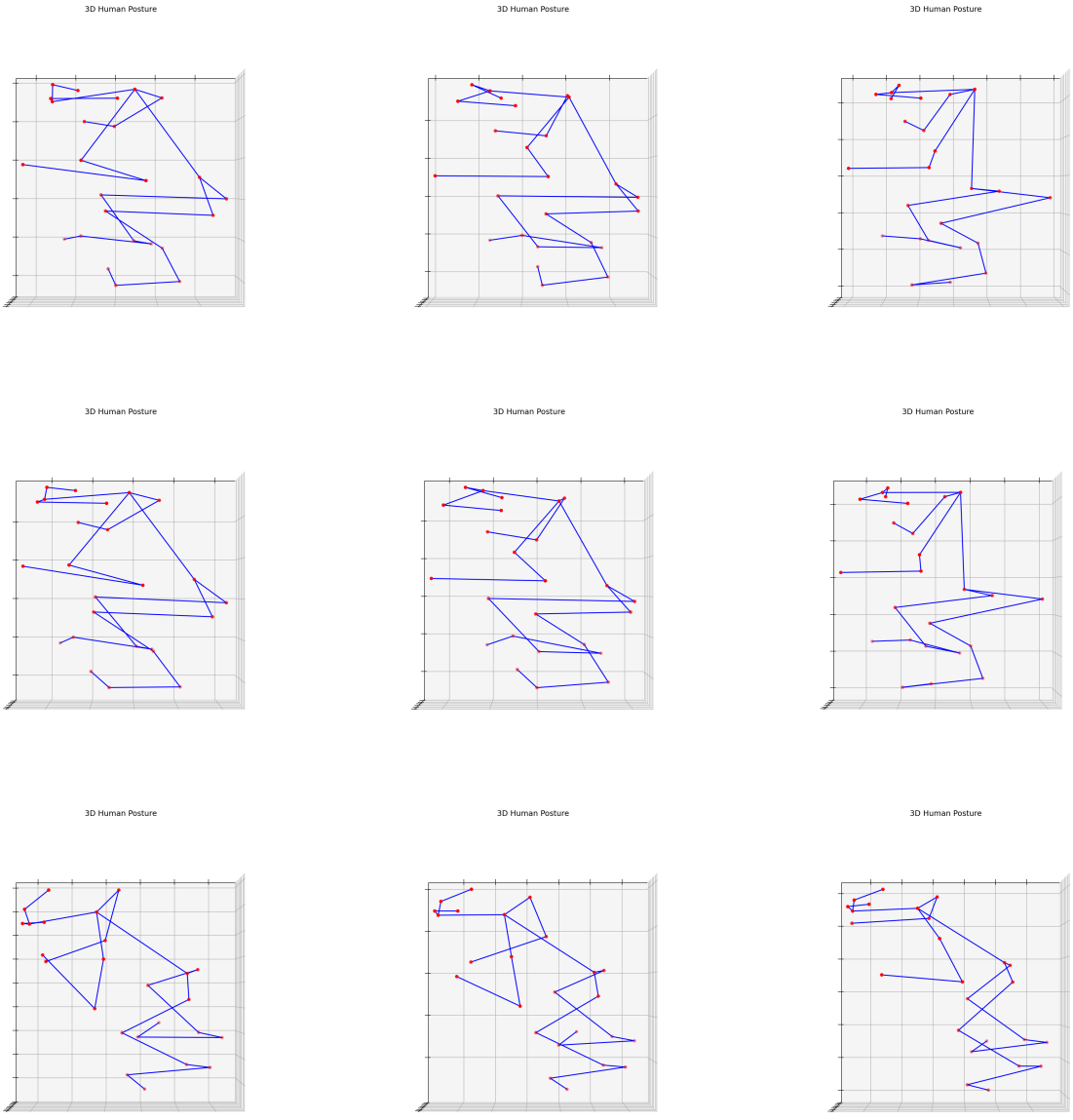
The first, second, and third columns represent the average error for the first, middle, and last predicted frames, respectively. The first, second, third, and fourth rows correspond to the error when one, two, three, and four persons were occluded.

Note that, by analyzing the values in Equation 7.3, we observe only positive values, indicating that including more people in the scene improves the reconstruction when compared to using only the target person. However, considering other reconstruction examples—such as those shown in Figures 35, 36, and 37—one can see that the first and second rows are visually very similar.

Therefore, even though the quantitative results in Equation 7.3 show improvements when all individuals are considered, the qualitative results suggest that these improvements are not substantially effective.

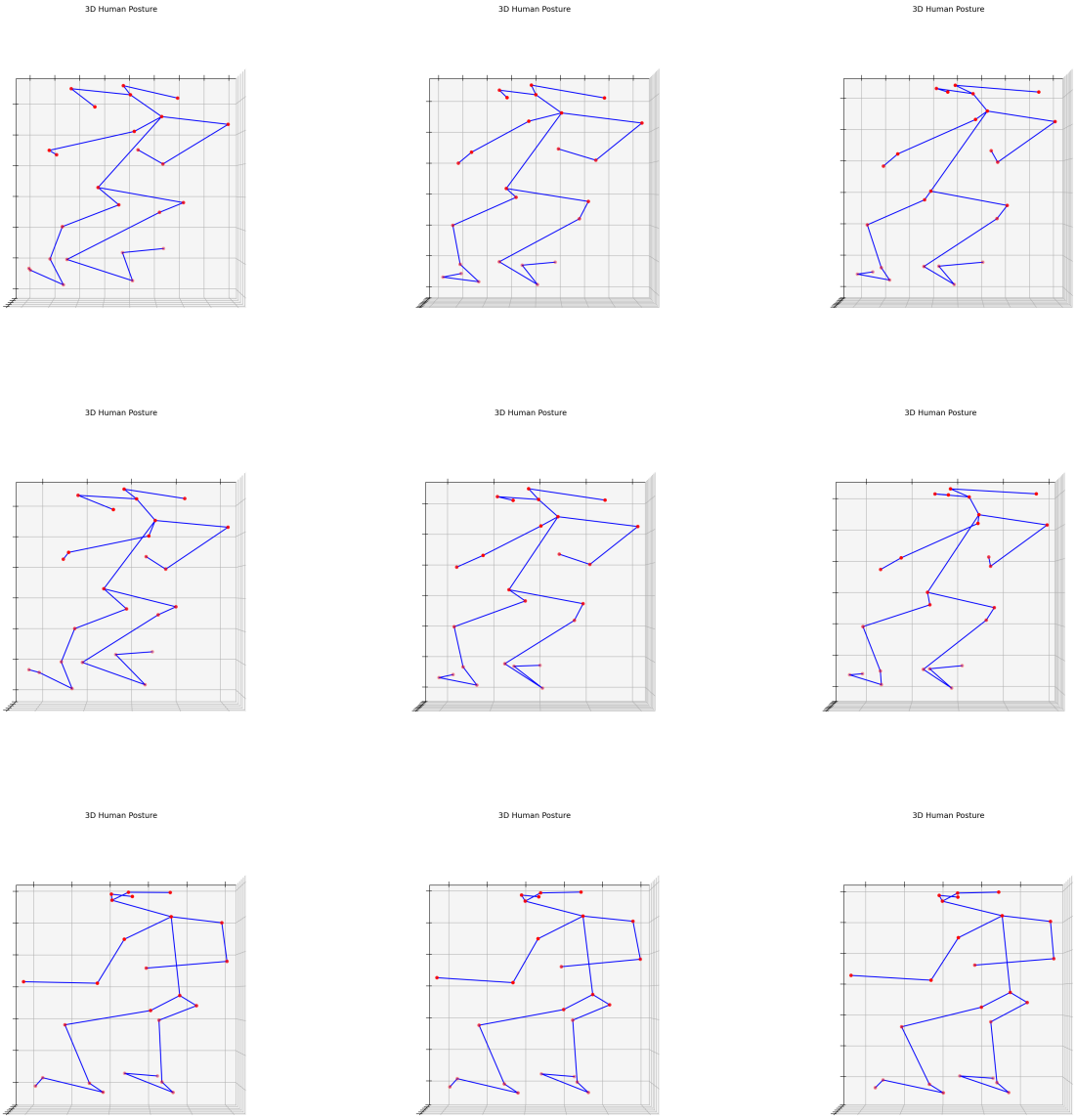
In this way, this chapter shows that although the method proposed in the previous chapters improved robbery detection performance, the experiments conducted in this chapter demonstrate that it has limited capacity to extract interpersonal dynamics from the scene.

Figure 35 – Scene of Figure 33b. Line 1: Reconstruction of the target person considering all persons in the scene. Line 2: Reconstruction of the target person disregarding all other persons in the scene. Line 3: Ground truth for the target person.



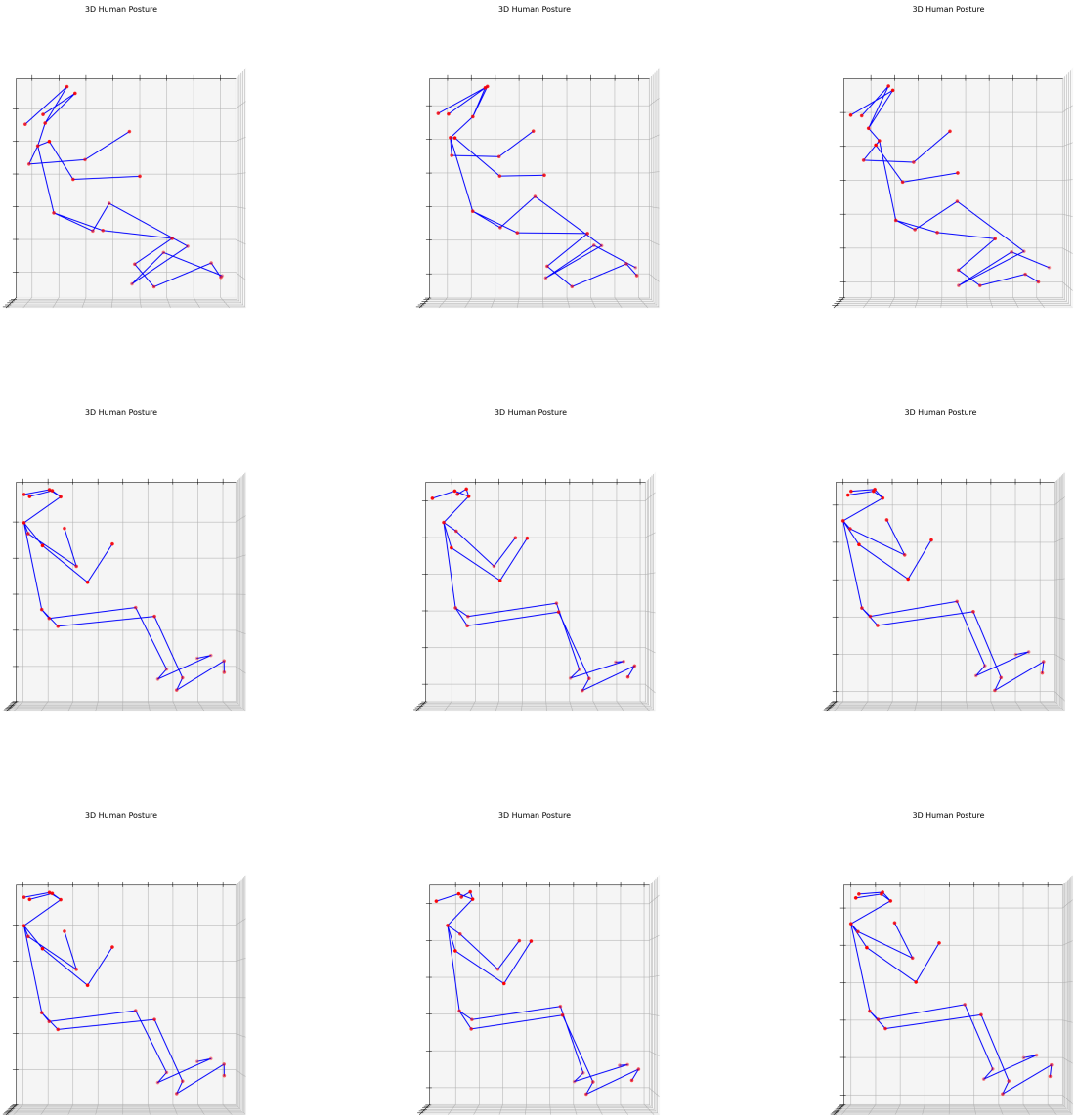
Source: Author's own work.

Figure 36 – Scene of Figure 33c. Line 1: Reconstruction of the target person considering all persons in the scene. Line 2: Reconstruction of the target person disregarding all other persons in the scene. Line 3: Ground truth for the target person.



Source: Author's own work.

Figure 37 – Scene of Figure 33d. Line 1: Reconstruction of the target person considering all persons in the scene. Line 2: Reconstruction of the target person disregarding all other persons in the scene. Line 3: Ground truth for the target person.



Source: Author's own work.

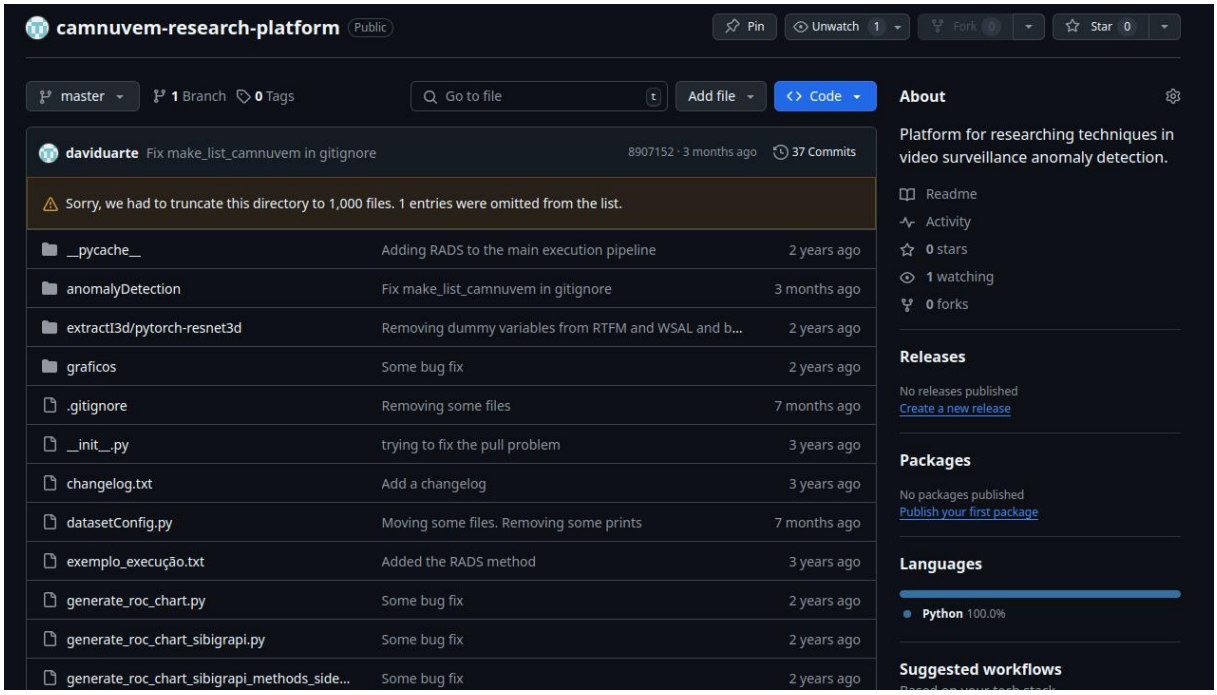
8 Conclusions

At the beginning of this thesis, the main hypothesis was presented: Is it possible to improve video surveillance robbery detection by extracting interpersonal relationships? The chosen approach to evaluate this hypothesis was the use of Self-Supervised Learning, due to its ability to extract visual representations without the need for human-made labels, and the use of human pose rather than raw video pixels, as it directly represents the human form and is cheaper to use for training models compared to raw pixels. To test this hypothesis, in Chapter 5, we first developed a Self-Supervised Learning (SSL) pretext task to estimate a person's feature vector and position. The idea behind this approach is that the model could learn to extract the interpersonal dynamics among people in the video. The results were evaluated using two datasets from the weakly supervised video anomaly detection task. Out of 12 experimental cases, 7 (58%) achieved good results. Although this method did not perform well in all scenarios, promising outcomes were observed for robbery anomalies in the CamNuvem dataset, achieving an AUC of 88.4%. This represents a 2.6% improvement compared to the baseline model without the proposed method. It is worth noting that 88.4% is a strong metric, indicating potential for real-world deployment after further optimizations. In relation to the initial hypothesis, this work supports its acceptance by demonstrating a notable improvement in the AUC for robbery detection.

This work also resulted in a tool (depiced on Figure 38) for executing multiple Multiple Instance Learning-based anomaly detection methods, which is freely available on GitHub [1]. This tool was developed in Python, using several libraries, such as NumPy, Scikit-Learn, and PyTorch. The execution instructions are also available on GitHub.

[1] <https://github.com/daviduarte/camnuvem-research-platform>

Figure 38 – GitHub repository of the tools developed in this work.



Source: Author's own work.

We also developed a second SSL pretext task that estimates the relationships between individuals by predicting a person's future position based on the dynamics of other people in the scene. The results were evaluated using two datasets from the supervised video anomaly detection task (a different task from the previous experiment). In the case where our method achieved the greatest improvement over the baseline, the AUC increased by 4.01%. The best AUC performance was obtained by our method in the robbery detection scenario using the CamNuvem dataset, reaching 75.81%, 1.54% AUC improvement compared to the same model pre-trained on the Kinetics-400 dataset. Although the improvement margin was modest, our method used significantly less data for pre-training and did not rely on any human-labeled annotations, unlike the Kinetics-400 approach. This demonstrates the effectiveness of our SSL pre-training strategy. Regarding our initial hypothesis, this experiment also supports its acceptance, as it produced better results than baseline methods while using much less data.

Despite the performance improvements reported in the aforementioned works, it remains inconclusive whether the proposed approach truly extracts interpersonal dynamics from the video or merely tracks the target person and learns general posture patterns. To address this question, in Chapter 7, we presented an analysis aimed at specifically verifying whether our approach captures interpersonal dynamics. The quantitative results support the hypothesis; however, the

qualitative results depict a rather uncertain scenario, indicating that the method did not contribute significantly. Therefore, we conclude that, in the evaluated scenarios, the main hypothesis of this work cannot be fully accepted. Further experiments are needed to validate its effectiveness in different contexts.

However, even if our method does not explicitly capture interpersonal dynamics in the video, the results show that it still contributes to robbery detection in surveillance videos. Although working with human posture reduced the complexity of handling videos during the training phase, there remains a computational cost at inference time in real-world applications, as human poses must still be extracted from raw video pixels. One potential solution is to deploy the methods described here on hardware equipped with AI-specific processors, such as those found in many modern Digital Video Recorders (DVRs), enabling edge-based image processing. Another limitation is that this work was focused solely on exploring the central research question: demonstrating that extracting information on interpersonal dynamics using Self-Supervised Learning (SSL) and human posture contributes to robbery detection in videos. As a result, no specific optimizations were applied to enhance the performance of the robbery detection models.

Overall, the proposed self-supervised learning approaches show strong potential to significantly improve anomaly detection in video surveillance systems, thereby contributing to enhanced public safety and security. Further research is required to investigate the applicability of this approach to other video understanding scenarios, as well as to evaluate its effectiveness in real-time anomaly detection settings.

As discussed in Chapter 6, although the robbery detection system achieved promising results, relying on human posture introduces latency due to the preprocessing required for person tracking and pose extraction. To address this limitation, a potential solution is to design a multi-layered detection system that balances speed and accuracy. In such a system, the first layer could consist of a deep learning anomaly detection model that operates directly on raw video pixels, providing fast inference but with limited precision. A second layer—triggered only under certain conditions, such as when the first model exhibits elevated uncertainty or error—could then invoke a more accurate but slower robbery detection model based on human posture. This selective triggering mechanism would prevent the more computationally intensive model from running continuously, thus preserving real-time performance. Such an architecture offers a promising approach for achieving high-quality, real-time robbery detection in video surveillance systems.

Furthermore, recent advances in self-supervised learning and feature learning—such as the Joint Embedding Predictive Architecture (JEPA) and multi-step reasoning networks—could be incorporated, adapted, and evaluated for the task of robbery detection (LECUN, 2022).

Finally, the methods proposed in this thesis were designed to enhance robbery detection, but they could also be applied to other video analysis tasks, such as human action recognition. However, certain application domains may benefit even more from the extraction of interpersonal dynamics as explored in this work—particularly tactical analysis in sports (STEIN et al., 2017; WU et al., 2022). In team sports, actions often result from the coordinated efforts of multiple players, and modeling interpersonal dynamics could lead to more accurate predictions of game events. Similarly, player selection could be enhanced by considering a player's interactions with teammates, rather than relying solely on individual skills. It might be more valuable to recruit a player who demonstrates strong synergy with teammates (and potentially even acquire their key collaborators), rather than signing a highly skilled player who lacks effective connections within the team.

References

ADAM, A. et al. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 30, n. 3, p. 555–560, 2008.

AGRAWAL, P.; CARREIRA, J.; MALIK, J. Learning to see by moving. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 37–45.

AMRAEE, S. et al. Abnormal event detection in crowded scenes using one-class svm. *Signal, Image and Video Processing*, Springer, v. 12, n. 6, p. 1115–1123, 2018.

ASSRAN, M. et al. Self-supervised learning from images with a joint-embedding predictive architecture. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2023. p. 15619–15629.

BA, J. L.; KIROS, J. R.; HINTON, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

BARBALAU, A. et al. Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection. *Computer Vision and Image Understanding*, Elsevier, v. 229, p. 103656, 2023.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: *Proceedings of the Fifth Annual Workshop on Computational Learning theory*. [S.l.: s.n.], 1992. p. 144–152.

BRASIL. *Código Penal (Decreto-Lei nº 2.848, de 7 de dezembro de 1940)*. 1940. <https://www.planalto.gov.br/ccivil_03/decreto-lei/del2848compilado.htm>. Art. 157 – Define o crime de roubo.

BROWN, T. B. et al. *Language Models are Few-Shot Learners*. 2020. Disponível em: <https://arxiv.org/abs/2005.14165>.

BUADES, A.; COLL, B.; MOREL, J.-M. A non-local algorithm for image denoising. In: IEEE. *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.], 2005. v. 2, p. 60–65.

CARON, M. et al. Deep clustering for unsupervised learning of visual features. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 132–149.

CARON, M. et al. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, v. 33, p. 9912–9924, 2020.

CARON, M. et al. Emerging properties in self-supervised vision transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 9650–9660.

CARREIRA, J. et al. *A Short Note on the Kinetics-700 Human Action Dataset*. arXiv, 2019. Disponível em: <https://arxiv.org/abs/1907.06987>.

CARREIRA, J.; ZISSERMAN, A. Quo vadis, action recognition. *A new model and the kinetics dataset. CoRR, abs/1705.07750*, v. 2, n. 3, p. 1, 2017.

CHEN, D. et al. Anomaly detection in surveillance video based on bidirectional prediction. *Image and Vision Computing*, Elsevier, v. 98, p. 103915, 2020.

CHEN, D. et al. Nm-gan: Noise-modulated generative adversarial network for video anomaly detection. *Pattern Recognition*, v. 116, p. 107969, 2021. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320321001564>.

CHEN, T. et al. A simple framework for contrastive learning of visual representations. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2020. p. 1597–1607.

CHEN, X. et al. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

CHEN, X.; HE, K. Exploring simple siamese representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 15750–15758.

CHENG, M.; CAI, K.; LI, M. *RWF-2000: An Open Large Scale Video Database for Violence Detection*. 2020. Disponível em: <https://arxiv.org/abs/1911.05913>.

CHICCO, D. Siamese neural networks: An overview. In: ____. *Artificial Neural Networks*. New York, NY: Springer US, 2021. p. 73–94. ISBN 978-1-0716-0826-5. Disponível em: <https://doi.org/10.1007/978-1-0716-0826-5_3>.

CHU, W. et al. Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos. *IEEE Transactions on Multimedia*, IEEE, v. 21, n. 1, p. 246–255, 2018.

CONG, Y.; YUAN, J.; LIU, J. Sparse reconstruction cost for abnormal event detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2011. p. 3449–3456.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2009. p. 248–255.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. Disponível em: <https://arxiv.org/abs/1810.04805>.

DISTANCES, C. M. S. Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, v. 26, p. 2292–2300, 2013.

DOERSCH, C.; GUPTA, A.; EFROS, A. A. Unsupervised visual representation learning by context prediction. In: *Proceedings of the IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 1422–1430.

DONG, Z.; QIN, J.; WANG, Y. Multi-stream deep networks for person to person violence detection in videos. In: SPRINGER. *Pattern Recognition: 7th Chinese Conference, CCPR 2016, Chengdu, China, November 5-7, 2016, Proceedings, Part I 7*. [S.l.], 2016. p. 517–531.

DOSHI, K.; YILMAZ, Y. Continual learning for anomaly detection in surveillance videos. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2020. p. 254–255.

DOSHI, K.; YILMAZ, Y. Online anomaly detection in surveillance videos with asymptotic bound on false alarm rate. *Pattern Recognition*, v. 114, p. 107865, 2021. ISSN 0031-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0031320321000522>.

DOSHI, K.; YILMAZ, Y. Towards interpretable video anomaly detection. In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. [S.l.: s.n.], 2023. p. 2654–2663.

DOSOVITSKIY, A. et al. Carla: An open urban driving simulator. In: PMLR. *Conference on robot learning*. [S.l.], 2017. p. 1–16.

DU, W.; FANG, M.; SHEN, M. *Siamese convolutional neural networks for authorship verification*. 2017. <https://cs231n.stanford.edu/reports/2017/pdfs/801.pdf>. Accessed: 2025-05-02.

DUMAN, E.; ERDEM, O. A. Anomaly detection in videos using optical flow and convolutional autoencoder. *IEEE Access*, IEEE, v. 7, p. 183914–183923, 2019.

FAN, Y. et al. Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder. *Computer Vision and Image Understanding*, v. 195, p. 102920, 2020. ISSN 1077-3142. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1077314218302674>.

FAN, Z. et al. Real-time and accurate abnormal behavior detection in videos. *Machine Vision and Applications*, Springer, v. 31, n. 7, p. 1–13, 2020.

FENG, J.; LIANG, Y.; LI, L. Anomaly detection in videos using two-stream autoencoder with post hoc interpretability. *Computational Intelligence and Neuroscience*, Hindawi, v. 2021, 2021.

GARCIA-COBO, G.; SANMIGUEL, J. C. Human skeletons and change detection for efficient violence detection in surveillance videos. *Computer Vision and Image Understanding*, v. 233, p. 103739, 2023. ISSN 1077-3142. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1077314223001194>.

GARRIDO, Q. et al. Learning and leveraging world models in visual representation learning. *arXiv preprint arXiv:2403.00504*, 2024.

GIDARIS, S.; SINGH, P.; KOMODAKIS, N. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

GOEL, S. et al. Humans in 4d: Reconstructing and tracking humans with transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2023. p. 14783–14794.

GOODFELLOW, I. et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*, v. 27, 2014.

GRILL, J.-B. et al. *Bootstrap your own latent: A new approach to self-supervised Learning*. 2020. Disponível em: <https://arxiv.org/abs/2006.07733>.

GU, C. et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 6047–6056.

HANSON, A. et al. Bidirectional convolutional lstm for the detection of violence in videos. In: *Proceedings of the European Conference on Computer Vision (ECCV) workshops*. [S.l.: s.n.], 2018. p. 0–0.

HASAN, M. et al. Learning temporal regularity in video sequences. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 733–742.

HE, K. et al. Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 9729–9738.

HEARST, M. et al. Support vector machines. *IEEE Intelligent Systems and their Applications*, v. 13, n. 4, p. 18–28, 1998.

HINAMI, R.; MEI, T.; SATOH, S. *Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge*. 2017. Disponível em: <https://arxiv.org/abs/1709.09121>.

HU, J. et al. Detecting anomalous events from unlabeled videos via temporal masked auto-encoding. In: *2022 IEEE International Conference on Multimedia and Expo (ICME)*. [S.l.: s.n.], 2022. p. 1–6.

HUANG, C. et al. Self-supervision-augmented deep autoencoder for unsupervised visual anomaly detection. *IEEE Transactions on Cybernetics*, IEEE, v. 52, n. 12, p. 13834–13847, 2021.

IBGE, Coordenação de Pesquisas por Amostra de Domicílios. *Vitimização : furto e roubo 2021*. 2022. Disponível em: <https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101983>.

IBGE, Coordenação de Pesquisas por Amostra de Domicílios. *Vitimização : sensação de segurança 2021*. 2022. Disponível em: <https://biblioteca.ibge.gov.br/index.php/biblioteca-catalogo?view=detalhes&id=2101984>.

JIANG, F. et al. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, Elsevier, v. 115, n. 3, p. 323–333, 2011.

JIANG, H. et al. Self-supervised relative depth learning for urban scene understanding. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 19–35.

JING, L.; TIAN, Y. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, 2020.

JING, L. et al. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018.

KAY, W. et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

KHAN, M. U. K.; PARK, H.-S.; KYUNG, C.-M. Rejecting motion outliers for efficient crowd anomaly detection. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 14, n. 2, p. 541–556, 2018.

KHANAM, R.; HUSSAIN, M. *What is YOLOv5: A deep look into the internal features of the popular object detector*. 2024. Disponível em: <https://arxiv.org/abs/2407.20892>.

KIM, D.; CHO, D.; KWEON, I. S. Self-supervised video representation learning with space-time cubic puzzles. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2019. v. 33, n. 01, p. 8545–8552.

KOCABAS, M.; KARAGOZ, S.; AKBAS, E. Self-supervised learning of 3d human pose using multi-view geometry. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 1077–1086.

KORBAR, B.; TRAN, D.; TORRESANI, L. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

LECUN, Y. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, v. 62, n. 1, p. 1–62, 2022.

LEDIG, C. et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4681–4690.

LI, D. et al. Unsupervised visual representation learning by graph-based consistent constraints. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2016. p. 678–694.

LI, W.; MAHADEVAN, V.; VASCONCELOS, N. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 36, n. 1, p. 18–32, 2013.

LIU, Y. et al. Collaborative normality learning framework for weakly supervised video anomaly detection. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 69, n. 5, p. 2508–2512, 2022.

LIZNERSKI, P. et al. Explainable deep one-class classification. *arXiv preprint arXiv:2007.01760*, 2020.

LOPER, M. et al. Smpl: A skinned multi-person linear model. In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. [S.l.: s.n.], 2023. p. 851–866.

LU, C. et al. Visual relationship detection with language priors. In: SPRINGER. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. [S.l.], 2016. p. 852–869.

LU, C.; SHI, J.; JIA, J. Abnormal event detection at 150 fps in matlab. In: *IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 2720–2727.

LUDWIG, K. et al. Self-supervised learning for human pose estimation in sports. In: IEEE. *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. [S.l.], 2021. p. 1–6.

LUO, W.; LIU, W.; GAO, S. Remembering history with convolutional lstm for anomaly detection. In: IEEE. *2017 IEEE International Conference on Multimedia and Expo (ICME)*. [S.l.], 2017. p. 439–444.

LV, H. et al. Localizing anomalies from weakly-labeled videos. *IEEE Transactions on Image Processing*, IEEE, v. 30, p. 4505–4515, 2021.

MA, D.; WANG, Q.; YUAN, Y. Anomaly detection in crowd scene via online learning. In: *Proceedings of International Conference on Internet Multimedia Computing and Service*. [S.l.: s.n.], 2014. p. 158–162.

MADAN, N. et al. Self-supervised masked convolutional transformer block for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), v. 46, n. 1, p. 525–542, jan. 2024. ISSN 1939-3539. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2023.3322604>.

MAQSOOD, R. et al. Anomaly recognition from surveillance videos using 3d convolution neural network. *Multimedia Tools and Applications*, Springer, v. 80, n. 12, p. 18693–18716, 2021.

MATHIEU, M.; COUPRIE, C.; LECUN, Y. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

MISRA, I.; ZITNICK, C. L.; HEBERT, M. Unsupervised learning using sequential verification for action recognition. *arXiv preprint arXiv:1603.08561*, Mar, v. 2, n. 7, p. 8, 2016.

MITCHELL, T. M. The need for biases in learning generalizations. Citeseer, 1980.

MONDAL, R.; CHANDA, B. Anomaly detection using context dependent optical flow. In: *Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing*. [S.l.: s.n.], 2018. p. 1–8.

MUNDHENK, T. N.; HO, D.; CHEN, B. Y. Improvements to context based self-supervised learning. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 9339–9348.

OQUAB, M. et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

ÖZTÜRK, H. İ.; CAN, A. B. Adnet: Temporal anomaly detection in surveillance videos. In: SPRINGER. *International Conference on Pattern Recognition*. [S.l.], 2021. p. 88–101.

PANG, G. et al. *Self-trained Deep Ordinal Regression for End-to-End Video Anomaly Detection*. 2020. Disponível em: <https://arxiv.org/abs/2003.06780>.

PATHAK, D. et al. Learning features by watching objects move. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 2701–2710.

PATHAK, D. et al. Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2536–2544.

PAULA, D. D. de; SALVADEO, D. H.; ARAUJO, D. M. de. Camnuvem: a robbery dataset for video anomaly detection. *Sensors*, MDPI, v. 22, n. 24, p. 10016, 2022.

PAULA, D. D. de et al. Self-supervised feature extraction for video surveillance anomaly detection. In: *2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. [S.l.: s.n.], 2023. p. 115–120.

PEIXOTO, B. et al. Multimodal violence detection in videos. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.: s.n.], 2020. p. 2957–2961.

PEREZ, M.; KOT, A. C.; ROCHA, A. Detection of real-world fights in surveillance videos. In: IEEE. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2019. p. 2662–2666.

RADFORD, A. et al. Language models are unsupervised multitask learners. *OpenAI blog*, v. 1, n. 8, p. 9, 2019.

RAGHAVENDRA, R.; BUE, A.; CRISTANI, M. *Unusual crowd activity dataset of University of Minnesota*. 2006. <http://mha.cs.umn.edu/proj_events.shtml#crowd>. [Online; accessed 19-October-2022].

RAJASEGARAN, J. et al. Tracking people with 3d representations. *CoRR*, abs/2111.07868, 2021. Disponível em: <https://arxiv.org/abs/2111.07868>.

RAJASEGARAN, J. et al. Tracking people by predicting 3d appearance, location and pose. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2022. p. 2740–2749.

RAJASEGARAN, J. et al. *On the Benefits of 3D Pose and Tracking for Human Action Recognition*. 2023.

RAMCHANDRAN, A.; SANGAIAH, A. K. Unsupervised deep learning system for local anomaly event detection in crowded scenes. *Multimedia Tools and Applications*, Springer, v. 79, n. 47, p. 35275–35295, 2020.

REN, Z.; LEE, Y. J. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 762–771.

REZAEI, F.; YAZDI, M. A new semantic and statistical distance-based anomaly detection in crowd video surveillance. *Wireless Communications and Mobile Computing*, Hindawi, v. 2021, 2021.

RISTEA, N.-C. et al. Self-distilled masked auto-encoders are efficient video anomaly detectors. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2024. p. 15984–15995.

RISTEA, N.-C. et al. *Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection*. 2022. Disponível em: <https://arxiv.org/abs/2111.09099>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.

ROSASCO, L. et al. Are loss functions all the same? *Neural computation*, MIT Press, v. 16, n. 5, p. 1063–1076, 2004.

RUFF, L. et al. Deep one-class classification. In: PMLR. *International Conference on Machine Learning*. [S.l.], 2018. p. 4393–4402.

SABOKROU, M. et al. *Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes*. 2017. Disponível em: <https://arxiv.org/abs/1609.00866>.

SABOKROU, M. et al. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. *IEEE Transactions on Image Processing*, v. 26, n. 4, p. 1992–2004, 2017.

SABOKROU, M. et al. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, Elsevier, v. 172, p. 88–97, 2018.

SABOKROU, M. et al. *Adversarially Learned One-Class Classifier for Novelty Detection*. 2018. Disponível em: <https://arxiv.org/abs/1802.09088>.

SABOKROU, M. et al. *AVID: Adversarial Visual Irregularity Detection*. 2018. Disponível em: <https://arxiv.org/abs/1805.09521>.

SAYED, N.; BRATTOLI, B.; OMMER, B. Cross and learn: Cross-modal self-supervision. In: SPRINGER. *German Conference on Pattern Recognition*. [S.l.], 2018. p. 228–243.

SCHÖLKOPF, B. et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, v. 13, n. 7, p. 1443–1471, 2001.

SHAH, S. et al. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: SPRINGER. *Field and service robotics*. [S.l.], 2018. p. 621–635.

SHANG, Y.; WU, X.; LIU, R. Multimodal violent video recognition based on mutual distillation. In: YU, S. et al. (Ed.). *Pattern Recognition and Computer Vision*. Cham: Springer Nature Switzerland, 2022. p. 623–637. ISBN 978-3-031-18913-5.

SHI, C. et al. Video anomaly detection via sequentially learning multiple pretext tasks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2023. p. 10330–10340.

SINGH, A.; JONES, M. J.; LEARNED-MILLER, E. G. Eval: Explainable video anomaly localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2023. p. 18717–18726.

SINGH, A.; PATIL, D.; OMKAR, S. *Eye in the Sky: Real-time Drone Surveillance System (DSS) for Violent Individuals Identification using ScatterNet Hybrid Deep Learning Network*. 2018. Disponível em: <https://arxiv.org/abs/1806.00746>.

SINGH, P.; PANKAJAKSHAN, V. A deep learning based technique for anomaly detection in surveillance videos. In: IEEE. *2018 Twenty Fourth National Conference on Communications (NCC)*. [S.l.], 2018. p. 1–6.

SRIVASTAVA, N.; MANSIMOV, E.; SALAKHUDINOV, R. Unsupervised learning of video representations using lstms. In: PMLR. *International conference on machine learning*. [S.l.], 2015. p. 843–852.

STEIN, M. et al. Bring it to the pitch: Combining video and movement data to enhance team sport analysis. *IEEE Transactions on Visualization and Computer Graphics*, IEEE, v. 24, n. 1, p. 13–22, 2017.

STIVAL, L. *Análise preditiva de fatores que influenciam na entrada na zona de finalização em partidas de futebol*. Universidade Estadual de Campinas, Faculdade de Tecnologia, 2022. Disponível em: <https://repositorio.unicamp.br/acervo/detalhe/1238666>.

SU, Y. et al. Human interaction learning on 3d skeleton point clouds for video violence recognition. In: VEDALDI, A. et al. (Ed.). *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020. p. 74–90. ISBN 978-3-030-58548-8.

SUDHAKARAN, S.; LANZ, O. *Learning to Detect Violent Videos using Convolutional Long Short-Term Memory*. 2017. Disponível em: <https://arxiv.org/abs/1709.06531>.

SULTANI, W.; CHEN, C.; SHAH, M. Real-world anomaly detection in surveillance videos. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 6479–6488.

SUN, C. et al. Adversarial 3d convolutional auto-encoder for abnormal event detection in videos. *IEEE Transactions on Multimedia*, v. 23, p. 3292–3305, 2021.

SUN, J.; SHAO, J.; HE, C. Abnormal event detection for video surveillance using deep one-class learning. *Multimedia Tools and Applications*, Springer, v. 78, n. 3, p. 3633–3647, 2019.

TAN, Y. H.; CHAN, C. S. Phrase-based image caption generator with hierarchical lstm network. *Neurocomputing*, Elsevier, v. 333, p. 86–100, 2019.

TAX, D. M.; DUIN, R. P. Support vector data description. *Machine learning*, Springer, v. 54, p. 45–66, 2004.

TIAN, Y. et al. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In: *IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2021. p. 4975–4986.

TRAN, D. et al. Learning spatiotemporal features with 3d convolutional networks. In: *IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2015. p. 4489–4497.

TRAN, D. et al. Deep end2end voxel2voxel prediction. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2016. p. 17–24.

ULLAH, W. et al. An efficient anomaly recognition framework using an attention residual lstm in surveillance videos. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 21, n. 8, p. 2811, 2021.

VASWANI, A. et al. Attention is all you need. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2017. p. 5998–6008.

VELASTIN, S. A.; GÓMEZ-LIRA, D. A. People detection and pose classification inside a moving train using computer vision. In: SPRINGER. *International Visual Informatics Conference*. [S.l.], 2017. p. 319–330.

VILLEGAS, R. et al. Decomposing motion and content for natural video sequence prediction. *arXiv preprint arXiv:1706.08033*, 2017.

VONDRICK, C.; PIRSIAVASH, H.; TORRALBA, A. Generating videos with scene dynamics. *Advances in neural information processing systems*, v. 29, p. 613–621, 2016.

VONDRICK, C. et al. Tracking emerges by colorizing videos. In: *Proceedings of the European conference on computer vision (ECCV)*. [S.l.: s.n.], 2018. p. 391–408.

WANG, C.; YAO, Y.; YAO, H. Video anomaly detection method based on future frame prediction and attention mechanism. In: IEEE. *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. [S.l.], 2021. p. 0405–0407.

WANG, G. et al. Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2022. p. 494–511.

WANG, J.; CHERIAN, A. Gods: Generalized one-class discriminative subspaces for anomaly detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 8201–8211.

WANG, S. et al. Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection. In: *Proceedings of the 26th ACM international conference on Multimedia*. [S.l.: s.n.], 2018. p. 636–644.

WANG, T. et al. Aed-net: An abnormal event detection network. *Engineering*, Elsevier, v. 5, n. 5, p. 930–939, 2019.

WANG, T. et al. Abnormal event detection via the analysis of multi-frame optical flow information. *Frontiers of Computer Science*, Springer, v. 14, n. 2, p. 304–313, 2020.

WANG, X. et al. Non-local neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 7794–7803.

WU, C. et al. Video anomaly detection using pre-trained deep convolutional neural nets and context mining. In: IEEE. *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*. [S.l.], 2020. p. 1–8.

WU, D. et al. Sports video analysis on large-scale data. In: *European Conference on Computer Vision*. [S.l.: s.n.], 2022. p. 19–36.

WU, P. et al. Deep learning for video anomaly detection: A review. *arXiv preprint arXiv:2409.05383*, 2024.

XU, D. et al. Learning deep representations of appearance and motion for anomalous event detection. *arXiv preprint arXiv:1510.01553*, 2015.

XU, K.; SUN, T.; JIANG, X. Video anomaly detection and localization based on an adaptive intra-frame classification network. *IEEE Transactions on Multimedia*, IEEE, v. 22, n. 2, p. 394–406, 2019.

XU, Y. et al. Vitpose: Simple vision transformer baselines for human pose estimation. *Advances in neural information processing systems*, v. 35, p. 38571–38584, 2022.

YANG, Y. et al. *Follow the Rules: Reasoning for Video Anomaly Detection with Large Language Models*. 2024. Disponível em: <https://arxiv.org/abs/2407.10299>.

YANG, Z.; LIU, J.; WU, P. Bidirectional retrospective generation adversarial network for anomaly detection in videos. *IEEE Access*, IEEE, v. 9, p. 107842–107857, 2021.

YANG, Z.; LIU, J.; WU, P. *Text Prompt with Normality Guidance for Weakly Supervised Video Anomaly Detection*. 2024. Disponível em: <https://arxiv.org/abs/2404.08531>.

YU, F.; KOLTUN, V. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

ZAHEER, M. Z. et al. *Old is Gold: Redefining the Adversarially Learned One-Class Classifier Training Paradigm*. 2020. Disponível em: <https://arxiv.org/abs/2004.07657>.

ZAHEER, M. Z. et al. Generative cooperative learning for unsupervised video anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2022. p. 14744–14754.

ZHANG, C. et al. Exploiting completeness and uncertainty of pseudo labels for weakly supervised video anomaly detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2023. p. 16271–16280.

ZHANG, J.; QING, L.; MIAO, J. Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. In: *2019 IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2019. p. 4030–4034.

ZHANG, R.; ISOLA, P.; EFROS, A. A. Colorful image colorization. In: LEIBE, B. et al. (Ed.). *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016. p. 649–666. ISBN 978-3-319-46487-9.

ZHOU, P. et al. Violent interaction detection in video based on deep learning. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2017. v. 844, n. 1, p. 012044.

Appendix

APPENDIX A – Hyperparameters

Table 20 presents a comprehensive list of hyperparameters used in the experiments described in Section 6.1.3. Human poses from SMPL, 3D Joints, and SMPL + 3D Joints are passed through a Multi-Layer Perceptron (MLP) to standardize the embedding size before being input into the transformer. The output sizes of these MLPs are referred to as "SMPL MLP output size" and "3D Joints MLP output size" in Table 20. The transformer expects an input vector of size 512, which is the concatenation of the Appearance MLP, SMPL MLP, and 3D Joints MLP. As previously noted, the appearance vector is disregarded by filling it with zeros; however, this vector is still processed by the transformer. In the experiments, the selection of SMPL, 3D Joints, or SMPL + 3D Joints is managed by filling the respective vectors with zeros when necessary.

Table 20 – Training hyperparameters and their values.

Hyperparameter	**Value**
Depth	16
Heads	16
Head Dim	64
Transform Input	512
SMPL MLP output size	128
3D Joints MLP output size	128
Appearance MLP output size	256
Dropout Rate	10%
Pretext batch size	4
Downstream batch size	4
Pretext Learning Rate	0.1×10^{-3}
Downstream Learning Rate	0.7×10^{-4}
Solver	AdamW

A.1 Raw data

Raw data for the experiments conducted in Section 6.2.4.

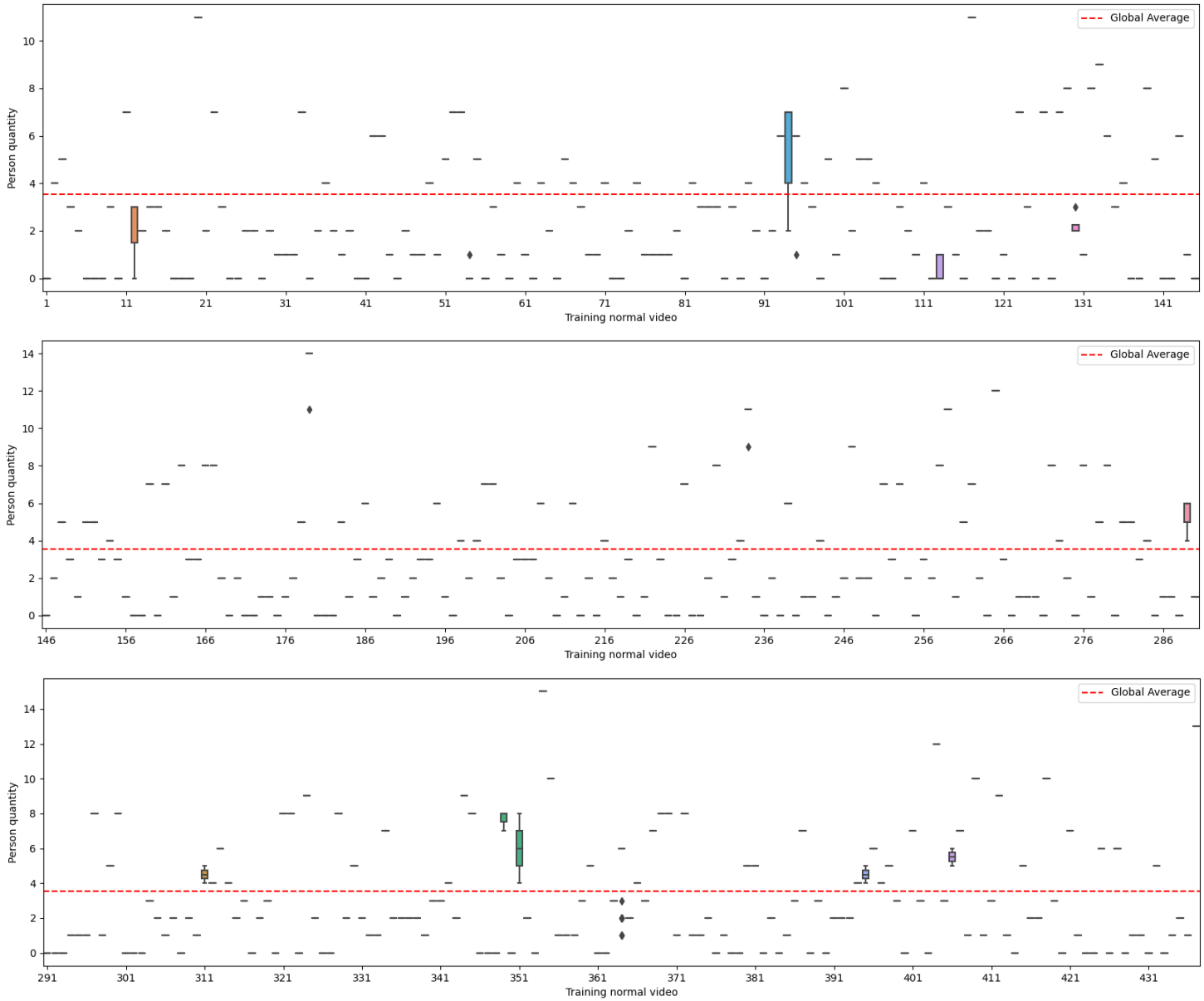
Table 21 – Raw data for Figure 32.

	Training set	**Validation set**
N=1	5,746	612
N=2	10,235	1,103
N=3	13,182	1,448
N=4	14,888	1,668
N=5	15,807	1,826

A.2 Person Quantity Analisys

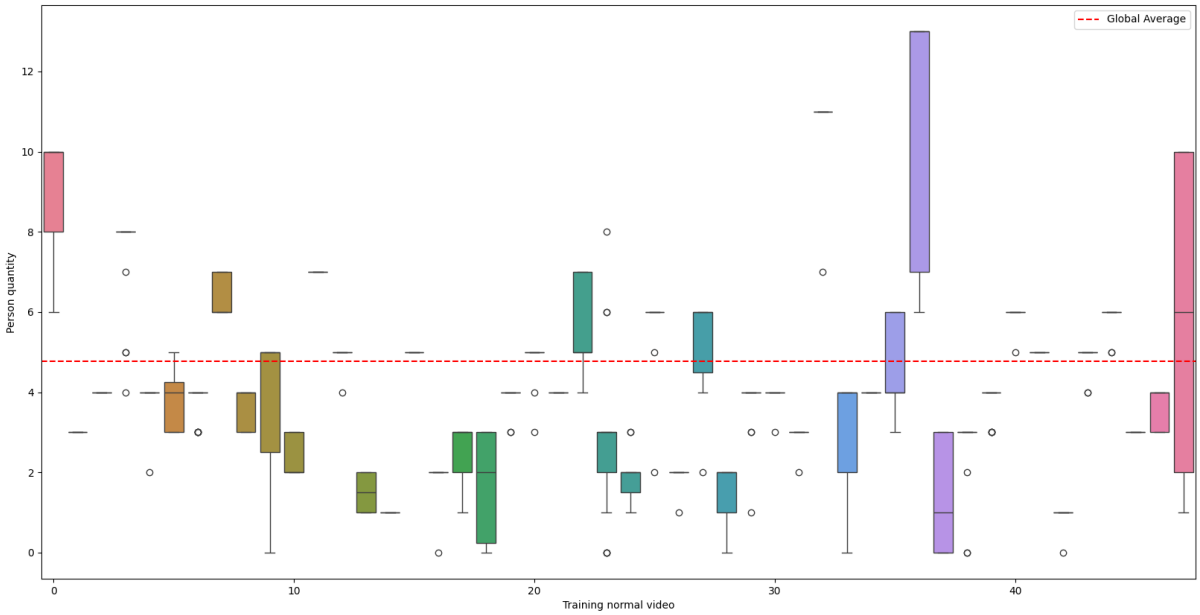
Average number of people detected in each video of the CamNuvem dataset. This figure is used to analyze the results discussed in Section 5.3.

Figure 39 – Detected person quantity (Y-axis) among all 437 normal training videos (X-axis). The horizontal red line specifies the number 3.53, which is the average number of detected persons across all scenes in all videos.



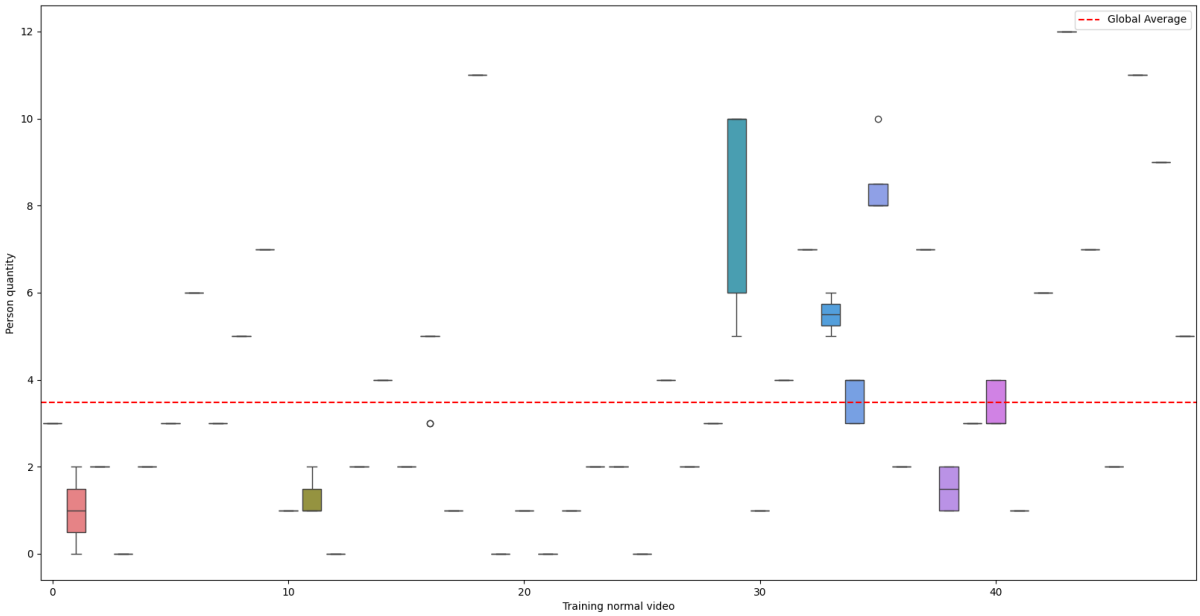
Source: Author's own work.

Figure 40 – Detected person quantity (Y-axis) among all 49 videos from the anomaly test set. The average number of detected persons across all scenes in all videos (red horizontal line) is 4.78.



Source: Author's own work.

Figure 41 – Detected person quantity (Y-axis) among all 49 videos from the normal test set. The average number of detected persons across all scenes in all videos (red horizontal line) is 3.47.



Source: Author's own work.

DADOS CURRICULARES	
IDENTIFICAÇÃO	
Nome	Davi Duarte de Paula
Data de nascimento	21/02/1996
Nacionalidade	Brasileira
Nome em citações bibliográficas	PAULA, D. D.; DE PAULA, D. D.; PAULA, DAVI; PAULA, DAVI D.; DE PAULA, DAVI D.
Currículo Lattes	http://lattes.cnpq.br/3111082564564736
ORCID	https://orcid.org/0000-0003-0230-2865
Google Scholar	https://scholar.google.com/citations?user=Dreue_sAAAAJ&hl=pt-BR
FORMAÇÃO ACADÊMICA	
2021-2025	Doutorado em Ciência da Computação. Instituição: Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP).
2018-2019	Mestrado em Ciência da Computação. Instituição: Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP).
2014-2017	Graduação em Ciência da Computação. Instituição: Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP).
PRODUÇÃO BIBLIOGRÁFICA	
Artigos em periódicos	
LÓPEZ VARGAS, JEAN PIERRE BRIK ; DE ABREU, KATARINY LIMA ; DUARTE DE PAULA, DAVI ; PINHEIRO SALVADEO, DENIS HENRIQUE ; ARANTES DE SOUZA, LILIAN FRANCISCO ; BÔA-VIAGEM RABELLO, CARLOS . Alternative Non-Destructive Approach for Estimating Morphometric Measurements of Chicken Eggs from Tomographic Images with Computer Vision. Foods, v. 13, p. 4039, 2024. DE ARAÚJO, DARLAN M. N. ; SALVADEO, DENIS H. P. ; DE PAULA, DAVI D. . Denoising digital breast tomosynthesis projections using deep learning with synthetic data as training set. JOURNAL OF MEDICAL IMAGING, v. 10, p. 034001, 2023. DE PAULA, DAVI D.; SALVADEO, DENIS H. P. ; DE ARAUJO, DARLAN M. N. . CamNuvem: A Robbery Dataset for Video Anomaly Detection. SENSORS, v. 22, p. 10016, 2022.	

Artigos em eventos

LÓPEZ VARGAS, JEAN PIERRE BRIK ; DUARTE DE PAULA, DAVI ; PINHEIRO SALVADEO, DENIS HENRIQUE ; BERGAMIM JÚNIOR, EMÍLIO . Generation of Simulated Dataset of Computed Tomography Images of Eggs and Extraction of Measurements Using Deep Learning. In: ESANN 2024, 2024, Bruges (Belgium) and online. ESANN 2024 proceesdings. Louvain-la-Neuve (Belgium): Ciaco - i6doc.com, 2024. p. 697.

DE PAULA, DAVI D.; SALVADEO, DENIS H. P. ; SILVA, LUCAS B. ; JUNIOR, UEMERSON P. . Self-Supervised Feature Extraction for Video Surveillance Anomaly Detection. In: 2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2023, Rio Grande. 2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2023. p. 115.

ARAUJO, D. M. N. ; SALVADEO, D. H. P. ; PAULA, DAVI D. . A benchmark of denoising Digital Breast Tomosynthesis in projection domain: neural network-based and traditional methods. In: Image Processing, 2022, San Diego, California - USA. SPIE Medical Imaging, 2022.

DE PAULA, DAVI; SALVADEO, DENIS ; DE ARAÚJO, DARLAN . Image reconstruction from projections of digital breast tomosynthesis using deep learning. In: Physics of Medical Imaging, 2021, Online Only. Medical Imaging 2021: Physics of Medical Imaging, 2021. p. 149.

NAKAMURA, DARLAN M. ; SALVADEO, DENIS H. ; PAULA, DAVI D. . Denoising digital breast tomosynthesis projections using convolutional neural networks. In: Image Processing, 2021, Online Only. Medical Imaging 2021: Image Processing, 2021. p. 53.

DE PAULA, DAVI D.; SALVADEO, DENIS H. P. . Breast Tomosynthesis Reconstruction Using Artificial Neural Networks with Deep Learning. In: 34th Conference on Graphics, Patterns and Images, 2021, Gramado, RS - Brazil. Anais Estendidos do XXXIV Conference on Graphics, Patterns and Images, 2021. p. 105-111.

CARVALHO, V. O. ; PAULA, D. D. ; PACHECO, M. V. ; SANTOS, W. E. ; PADUA, R. ; REZENDE, S. O. . Ranking Association Rules by Clustering through Interestingness. In: 16th Mexican International Conference on Artifial Intelligence, 2017, Ensenada, Baja California. Lecture Notes in Artificial Intelligence, 2017. p. 15.