Ain Shams University
Faculty of Computer & Information Sciences
Scientific Computing Department

Passenger Flow Tracking

Anomaly Detection / Smart Surveillance system

By:

Said Khaled Said Mohamed Ashour

Mahmoud Ali Mahmoud Esmail

Anas Ahmed Mohamed

Gamal Ahmed Abd El Hakm

Mohammed Zien El-abdine Abd-elaziem

Under Supervision of:

Dr. Dina El Sayad,
Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

TA. Marwa Shams,
Scientific Computing Department,
Faculty of Computer and Information Sciences,
Ain Shams University.

July 2023

Acknowledgments

All praise and thanks to ALLAH, who provided us with the ability to complete this work.

We are grateful to our family who are always providing help and support throughout the whole years of study. We hope we can return that favor to them.

We also offer our sincerest gratitude to our supervisors, Prof. *Dr. Dina El Sayad*, and *T.A. Marwa Shams* who have supported us throughout our thesis with their patience, knowledge, and experience.

We would like to thank our colleges who worked on this project, the project would not have been successful without their cooperation and input.

Finally, we would like to thank our friends and all the people who gave us support and encouragement.

Abstract

This thesis presents the development and implementation of a real-time anomaly surveillance system. The system leverages advanced techniques from computer vision, machine learning, and artificial intelligence to enable proactive monitoring and detection of anomalies in various environments. Through the utilization of state-of-the-art models and algorithms, the system provides real-time analysis of multiple camera feeds, automatically detects anomalies, and triggers appropriate actions for timely response. The system offers a user-friendly interface, customizable settings, and integration with email notifications for efficient anomaly management. Extensive experiments and evaluations demonstrate the effectiveness and practicality of the system in enhancing situational awareness and improving overall security.

ملخص

تقدم هذه الأطروحة تطوير وتنفيذ نظام مراقبة آني للظواهر الغريبة. يستفيد النظام من التقنيات المتقدمة من رؤية الحاسوب والتعلم الآلي والذكاء الاصطناعي لتمكين المراقبة الاستباقية والكشف عن الظواهر الغريبة في البيئات المختلفة. من خلال استخدام أحدث النماذج والخوارزميات، يوفر النظام تحليلًا في الوقت الفعلي لتغذية كاميرات متعددة، ويكتشف تلقائيًا الظواهر الغريبة، ويطلق الإجراءات المناسبة للاستجابة في الوقت المناسب. يوفر النظام واجهة سهلة الاستخدام، وإعدادات قابلة للتخصيص، وتكاملًا مع إشعارات البريد الإلكتروني لإدارة فعالة للظواهر الغريبة. تُظهر التجارب والتقييمات المكثفة فعالية النظام وعمليته في تعزيز الوعي بالأوضاع وتحسين الأمن العام.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Introduction

In today's rapidly evolving world, ensuring the safety and security of our surroundings is of paramount importance. As technological advancements continue to shape our lives, the need for effective surveillance systems that can quickly detect and respond to anomalies becomes increasingly crucial. Real-time anomaly surveillance systems have emerged as a powerful solution, providing proactive monitoring and alerting capabilities to mitigate potential threats and risks. This project focuses on developing an advanced real-time anomaly surveillance system that leverages state-of-the-art technologies to enhance situational awareness and enable timely responses to abnormal events.

1.2 Problem Definition

The problem at hand is the inherent challenge in efficiently monitoring large-scale environments and swiftly identifying anomalies that deviate from normal patterns or behaviors. Traditional surveillance systems often rely on manual monitoring, making it labor-intensive, time-consuming, and prone to human error. Moreover, detecting anomalies in real-time poses an additional challenge, as the sheer volume of data generated by multiple cameras requires intelligent processing algorithms to analyze and identify abnormal activities accurately. Thus, there is a need for an automated and intelligent system that can continuously monitor multiple camera feeds, detect anomalies in real-time, and provide timely alerts for proactive decision-making and response.

1.3 Motivation

The motivation behind this project stems from the pressing need for effective anomaly surveillance systems to bolster security measures in various domains. By harnessing the power of advanced computer vision, machine learning, and artificial intelligence techniques, we aim to develop a system that can autonomously monitor diverse environments, such as public spaces, transportation hubs, and critical infrastructure. Our goal is to enable early detection of anomalies, ranging from suspicious activities to potential safety hazards, thereby minimizing the risk of incidents, improving emergency response times, and ultimately enhancing overall safety and security for individuals and organizations alike. By developing an intelligent real-time anomaly surveillance system, we aim to contribute to the advancement of surveillance technologies and their practical implementation in real-world scenarios.

1.4 Document Organization

- Chapter 1: Introduction

This chapter introduces the real-time anomaly surveillance system. It presents the problem statement, highlights the motivation behind the project, and outlines the structure of the thesis.

- Chapter 2: Related Works

In this chapter, an overview of related works in the field of anomaly detection is provided. The chapter covers supervised learning, unsupervised learning, weakly-supervised learning, and multiple-instance learning techniques. Several notable algorithms and models used in anomaly detection are discussed.

- Chapter 3: System Architecture

Chapter 3 focuses on the system architecture of the real-time anomaly surveillance system. It introduces the RTFM (Real-Time Feature Mining) method and the I3D (Inflated 3D) model. The chapter explains the theoretical motivation behind the chosen methods and presents a data flow diagram depicting the system's architecture at different levels.

- Chapter 4: System Implementation and Results

This chapter delves into the implementation details of the system and provides a description of the software tools used. It covers feature extraction, data augmentation, anomaly detection algorithms, and various libraries and frameworks utilized in the development process. Additionally, the chapter discusses the hardware setup used and presents the experimental setup and the results obtained from the project.

- Chapter 5: Visual Walkthrough

Chapter 5 provides a visual walkthrough of the real-time anomaly surveillance system. It presents the different states of the system, including the initial state, the toolbar options, the normal state, the anomaly state, and the full program running. Furthermore, it highlights the process of saving anomaly screenshots in a designated folder and the functionality of sending notifications via email.

Chapter 2: Related Works

2.1 Introduction

Anomaly detection plays a crucial role in various domains, including cybersecurity, fraud detection, fault diagnosis, and surveillance systems. The ability to identify abnormal instances within a dataset is essential for maintaining system integrity, detecting suspicious activities, and ensuring operational efficiency. Over the years, researchers have developed numerous approaches and techniques for anomaly detection, each addressing the challenges posed by different data characteristics and application domains.

In this chapter, we explore the related works in anomaly detection, focusing on different learning paradigms and methodologies employed in the field. The chapter provides a comprehensive overview of supervised learning, unsupervised learning, weakly supervised learning, and multiple instance learning approaches for detecting anomalies.

First, we delve into the realm of supervised learning, where labeled training data is utilized to train models for anomaly detection. We discuss the use of handcrafted features and the adoption of deep neural networks to extract meaningful representations from data. The advantages and limitations of supervised learning in anomaly detection are explored, highlighting the need for labeled anomaly data and the potential challenges in detecting subtle anomalies.

Next, we delve into the realm of unsupervised learning, which aims to identify anomalies solely based on the characteristics of the data without any prior labeled information. We explore traditional unsupervised anomaly detection methods that utilize handcrafted features and statistical techniques to model normal data distribution and detect deviations. We also delve into recent advances that leverage deep learning techniques to enhance anomaly detection performance.

Moving on, we investigate weakly supervised learning approaches, which leverage limited labeled abnormal samples to improve anomaly detection. We discuss the incorporation of prior knowledge, statistical deviations, and end-to-end learning of anomaly scores through neural deviation learning. These approaches demonstrate improved data efficiency and enhanced anomaly scoring compared to unsupervised methods.

Lastly, we explore multiple-instance learning (MIL) as a framework for anomaly detection. MIL approaches operate at the bag level, where a bag represents a collection of instances. The labels are assigned to bags rather than individual instances, enabling the learning of discriminative patterns between normal and abnormal bags. We examine various MIL algorithms and their strengths in addressing the challenges posed by imprecise or weak instance-level labels.

By delving into these related works, this chapter aims to provide a comprehensive understanding of different approaches and methodologies employed in anomaly detection. The insights gained from these studies will help in identifying suitable techniques for detecting anomalies in various application domains, furthering the development of effective and efficient anomaly detection systems.

2.2 Supervised Learning

Supervised learning is a machine learning paradigm for problems where the available data consists of labeled examples, meaning that each data point contains features (covariates) and an associated label. The goal of supervised learning algorithms is learning a function that maps feature vectors (inputs) to labels (output), based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to

unseen situations in a “reasonable” way (see inductive bias). This statistical quality of an algorithm is measured through the so-called generalization error.

Steps to follow:

To solve a given problem of supervised learning, one must perform the following steps:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In the case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, an entire sentence of handwriting or perhaps a full paragraph of handwriting.
2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered, and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains several features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support-vector machines or decision trees.
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.
6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

Although this may yield the best results working with supervised learning on anomaly detection datasets is a very hard task as it requires frame level labeling which requires human efforts and lots of time.

Most anomaly detection datasets Like – ShanghaiTech which is a medium-scale data set from fixed angle street video surveillance. It has 13 different background scenes and 437 videos, including 307 normal videos and 130 anomaly video and UCF-Crime which is a large-scale anomaly detection data set that contains 1900 untrimmed videos with a total duration of 128 hours from real-world street and indoor surveillance cameras –

are unbalanced datasets, and even if the anomaly videos don't have all anomaly frames, it may have only 10% anomaly to 90% normal ratio which makes it a really hard task for a fully supervised approach.[1]

2.3 Unsupervised Learning

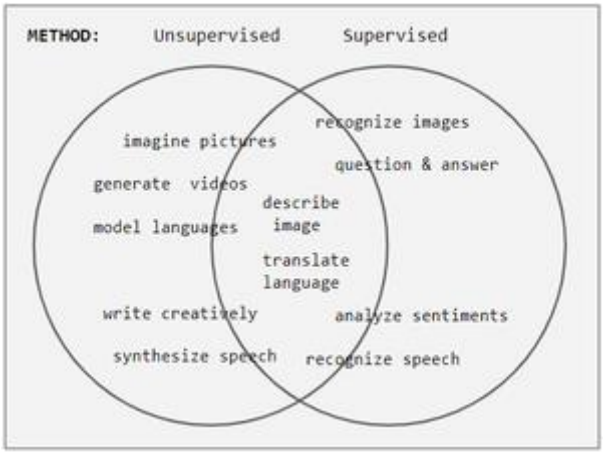
2.3.1 Introduction

Unsupervised learning refers to algorithms that learn patterns from unlabeled data.

In contrast to supervised learning where models learn to map the input to the target output (e.g., images labeled as a "cat" or "fish"), unsupervised methods learn concise representations of the input data, which can be used for data exploration or to analyze or generate new data. The other levels in the supervision spectrum are reinforcement learning where the machine is given only a "performance score" as guidance, and semi-supervised learning where only a portion of training data is labeled.

2.3.1.1 Neural Networks

Tasks vs. Methods:



Tendency for a task to employ supervised vs. unsupervised methods. Task names straddling circle boundaries are intentional. It shows that the classical division of imaginative tasks (left) employing unsupervised methods is blurred in today's learning schemes.

Neural network tasks are often categorized as discriminative (recognition) or generative (imagination). Often but not always, discriminative tasks use supervised methods and generative tasks use unsupervised (see Venn diagram); however, the separation is very hazy. For example, object recognition favors supervised learning, but unsupervised learning can also cluster objects into groups. Furthermore, as progress marches onward some tasks employ both methods, and some tasks swing from one to another. For example, image recognition started off as heavily supervised, but became hybrid by employing unsupervised pre-training, and then moved towards supervision again with the advent of dropout, ReLU, and adaptive learning rates.

Training*:*

During the learning phase, an unsupervised network tries to mimic the data it's given and uses the error in its mimicked output to correct itself (i.e., correct its weights and biases). Sometimes the error is expressed as a low probability that the erroneous output occurs, or it might be expressed as an unstable high energy state in the network.

In contrast to supervised methods' dominant use of backpropagation, unsupervised learning also employs other methods including: Hopfield learning rule, Boltzmann learning rule,

Contrastive Divergence, Wake Sleep, Variational Inference, Maximum Likelihood, Maximum A Posteriori, Gibbs Sampling, and backpropagating reconstruction errors or hidden state reparameterizations. See the table below for more details.[2]

2.3.2 Unsupervised Anomaly Detection

Traditional anomaly detection methods assume the availability of normal training data only and address the problem with one-class classification using handcrafted features.

2.3.2.1 Learning Fine-grained Image Similarity with Deep Ranking.[3]

The methodology used in this paper involves proposing a deep ranking model that employs deep learning techniques to learn similarity metrics directly from images. The model uses a novel multiscale network structure to describe the images effectively and an efficient triplet sampling algorithm to learn the model with distributed asynchronized stochastic gradient. The model is evaluated on a human-labeled dataset using triplets to label the similarity relationship of the images. The performance of the model is determined by the fraction of the triplet orderings that agrees with the ranking of the model. The experiments show that the proposed deep ranking model outperforms the hand-crafted visual feature-based approaches and deep classification models by a large margin.

With the advent of deep learning, more recent approaches use the features from pre-trained deep neural networks.

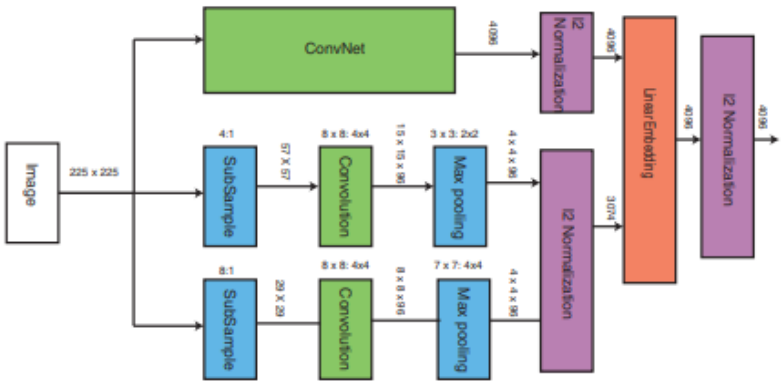


Figure 1: The multiscale network structure. Ech input image goes through three paths.

2.3.2.2 Unmasking the abnormal events in video. [4]

The proposed methodology uses a sliding window algorithm to label short-lasting events as abnormal if the amount of change from the immediately preceding event is substantially large. Motion and appearance features are extracted from the frames, and a binary classifier is trained with high regularization. The unmasking technique is then applied iteratively to the classifier to remove the most discriminant features, and higher training accuracy rates of the intermediately obtained classifiers represent abnormal events. The anomaly score for the last w frames is computed as the mean of the retained accuracy rates. The proposed method achieves state-of-the-art results in real-time.

2.3.2.3 Deep Ordinal Regression for Video Anomaly Detection. [5]

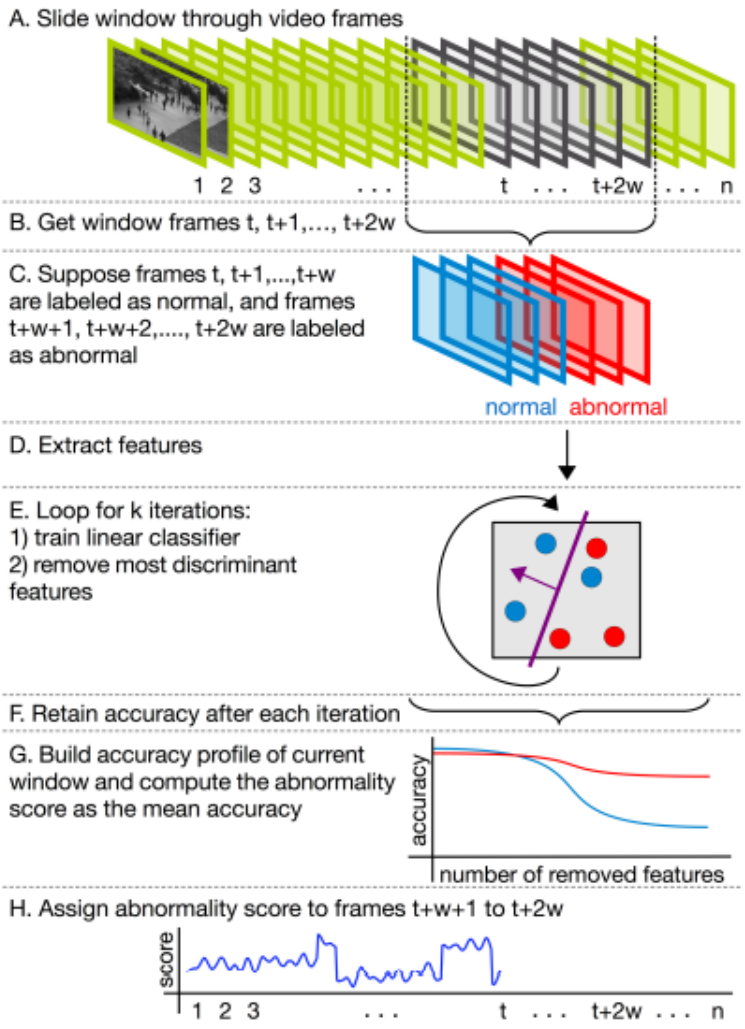


Figure 2: the anomaly detection framework based on unmasking. The steps are processed in sequential order from (A) to (H). Best viewed in color.

The proposed methodology is a self-training deep neural network for ordinal regression that enables joint representation learning and anomaly scoring without the need for manually labeled normal/abnormal data. The methodology consists of three major modules: (1) initial anomaly detection, (2) end-to-end anomaly scoring, and (3) iterative refinement of the membership of anomalous and normal frames. The initial anomaly detection module uses state-of-the-art unsupervised anomaly detection methods to obtain a set of frames that can be identified as belonging to anomalous and normal frames with high probability. These frames are then fed into the end-to-end anomaly scoring module to optimize the anomaly scores. The resulting anomaly scores are used to update the membership of anomalous and normal frames, which is then fed back into the end-to-end anomaly scoring module for further refinement. The proposed methodology outperforms state-of-the-art methods by a substantial margin and offers effective human-in-the-loop anomaly detection.

2.3.2.4 Exploring Self-attention for Image Recognition. [6]

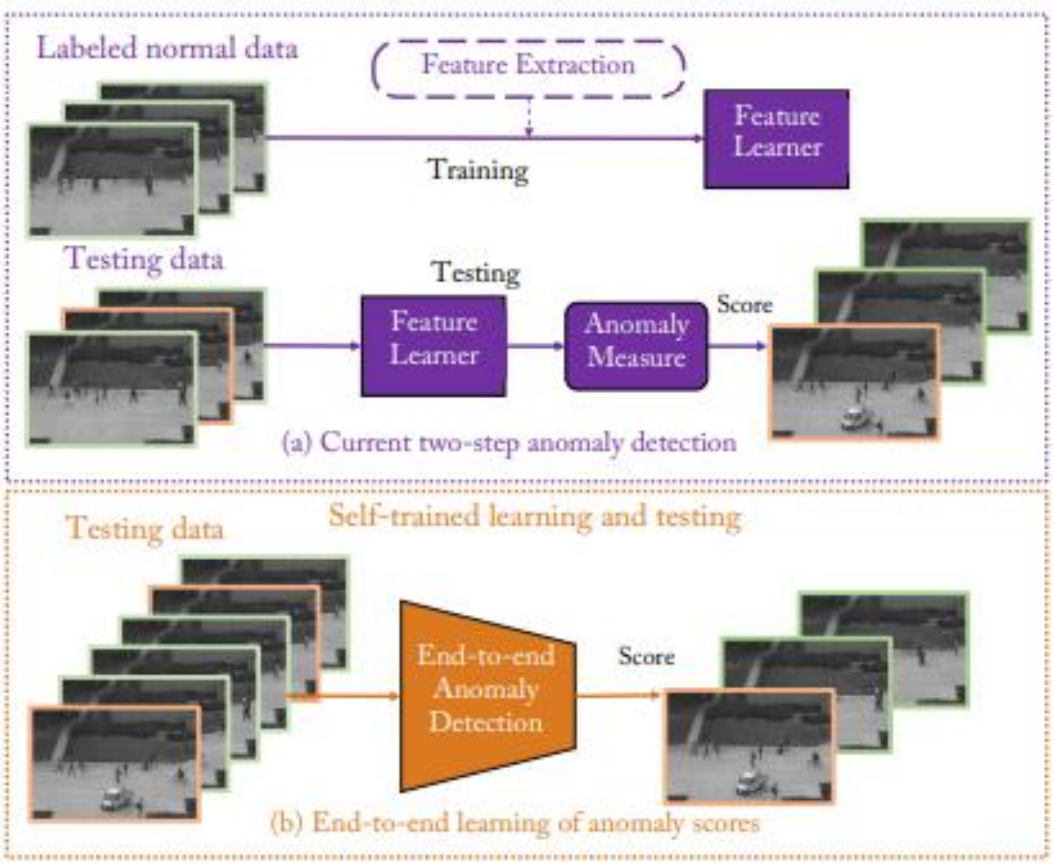


Figure 3: Pipelines of (a) two-step and (b) end-to-end anomaly detection.

The proposed methodology explores the use of self-attention networks for image recognition. The authors compare the effectiveness of self-attention networks to convolutional networks and investigate the potential benefits of self-attention networks in terms of robustness and generalization.

They conduct controlled experiments and ablation studies to analyze the performance of different self-attention configurations and compare them to convolutional networks.

The authors also release their full implementation and experimental setup open source to facilitate comparison and assist future work in this area.

Overall, the proposed methodology aims to provide insights into the effectiveness of self-attention networks for image recognition and their potential benefits over convolutional networks.

Table 1: Self-attention networks for image recognition.

Layers	Output Size	SAN10	SAN15	SAN19
Input	224×224×3	64-d linear		
Transition	112×112×64	2×2, stride 2 max pool → 64-d linear		
SA Block	112×112×64	[3×3, 16-d sa; 64-d linear] ×2	[3×3, 16-d sa; 64-d linear] ×3	[3×3, 16-d sa; 64-d linear] ×3
Transition	56×56×256	2×2, stride 2 max pool → 256-d linear		
SA Block	56×56×256	[7×7, 64-d sa; 256-d linear] ×1	[7×7, 64-d sa; 256-d linear] ×2	[7×7, 64-d sa; 256-d linear] ×3
Transition	28×28×512	2×2, stride 2 max pool → 512-d linear		
SA Block	28×28×512	[7×7, 128-d sa; 512-d linear] ×2	[7×7, 128-d sa; 512-d linear] ×3	[7×7, 128-d sa; 512-d linear] ×4
Transition	14×14×1024	2×2, stride 2 max pool → 1024-d linear		
SA Block	14×14×1024	[7×7, 256-d sa; 1024-d linear] ×4	[7×7, 256-d sa; 1024-d linear] ×5	[7×7, 256-d sa; 1024-d linear] ×6
Transition	7×7×2048	2×2, stride 2 max pool → 2048-d linear		
SA Block	7×7×2048	[7×7, 512-d sa; 2048-d linear] ×1	[7×7, 512-d sa; 2048-d linear] ×2	[7×7, 512-d sa; 2048-d linear] ×3
Classification	1×1×1000	global average pool → 1000-d linear → softmax		

Others apply constraints on the latent space of normal manifold to learn compact normality representations.

2.3.2.5 Classification-Based Anomaly Detection for General Data. [7]

The proposed method is called GOAD (Generalized Open-set Anomaly Detection). It is based on classification and uses normal training data only. The method transforms the data into M subspaces and learns a feature space such that inter-class separation is larger than intra-class separation. It then uses this criterion to determine if a new data point is normal or anomalous.

2.3.2.6 Anomaly Detection with Multi-scale Interpolated Gaussian Descriptors. [8]

The paper introduces a novel unsupervised anomaly detection and localization method that addresses two common issues in current systems. It proposes a robust distribution estimation technique using adversarially interpolated descriptors and a Gaussian classifier to handle under-represented classes of normal images. Additionally, a new anomaly identification criterion is proposed to accurately detect and localize multi-scale structural and non-structural anomalies. Extensive experiments on various datasets demonstrate that the proposed approach outperforms the current state-of-the-art methods in unsupervised anomaly detection and localization.

Alternatively, some approaches depend on data reconstruction using generative models to learn the representations of normal samples by (adversarially) minimizing the reconstruction error.

2.3.2.7 Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection. [9]

This paper proposes an approach called Memory-Augmented Autoencoder (MemAE) to address a limitation in using deep autoencoders for anomaly detection. While autoencoders are commonly used for this task by reconstructing anomalies with higher error than normal inputs, they sometimes “generalize” well enough to reconstruct anomalies accurately, leading to missed detections. To overcome this, the authors augment the autoencoder with a memory module. The MemAE retrieves relevant memory items based on the encoding from the encoder, and during training, the memory contents are updated to represent the prototypical elements of normal data. During testing, the learned memory is fixed, and reconstruction is obtained from selected memory records of normal data, which strengthens the reconstructed errors on anomalies for

detection. MemAE is versatile across different data types and shows excellent generalization and effectiveness in various experiments on different datasets.

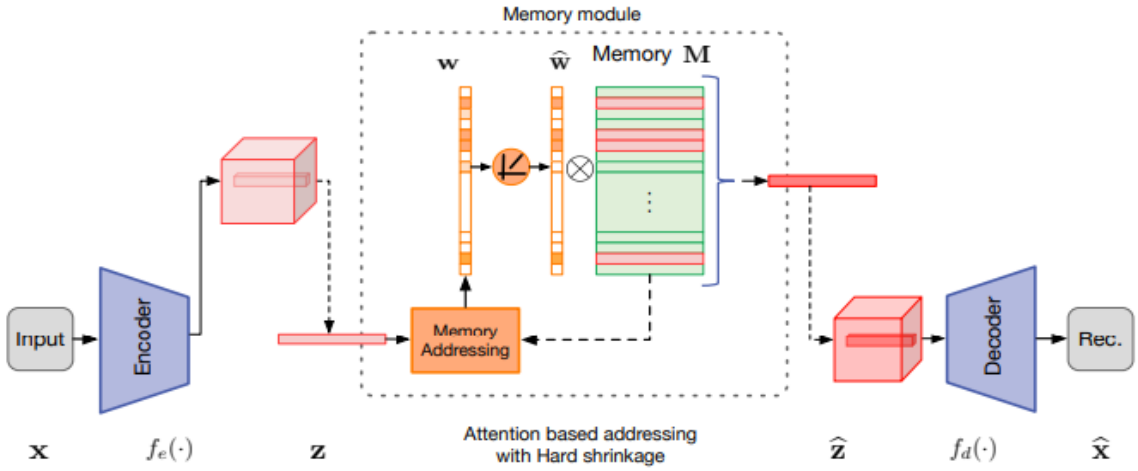


Figure 4: Diagram of the proposed MemAE.

2.3.2.8 Object-centric Auto-encoders and Dummy Anomalies for Abnormal Event Detection in Video. [10]

The paper discusses the challenging task of abnormal event detection in videos and highlights the limitations of existing approaches that rely on outlier detection due to the scarcity of anomalous training data. To address this, the authors propose a two-fold contribution. Firstly, they introduce an unsupervised feature learning framework based on object-centric convolutional auto-encoders that encode both motion and appearance information. This framework aids in capturing discriminative features for abnormal event detection. Secondly, they propose a supervised classification approach that clusters training samples into normality clusters and employs a one-versus-rest abnormal event classifier to separate each normality cluster from the rest, treating the other clusters as dummy anomalies. During inference, an object is labeled as abnormal if the highest classification score assigned by the one-versus-rest classifiers is negative. The proposed approach is evaluated on four benchmark datasets (Avenue, ShanghaiTech, UCSD, and UMN) and demonstrates superior results across all datasets. Particularly, on the large-scale ShanghaiTech dataset, the method achieves an absolute gain of 8.4% in terms of frame-level abnormal event detection accuracy.

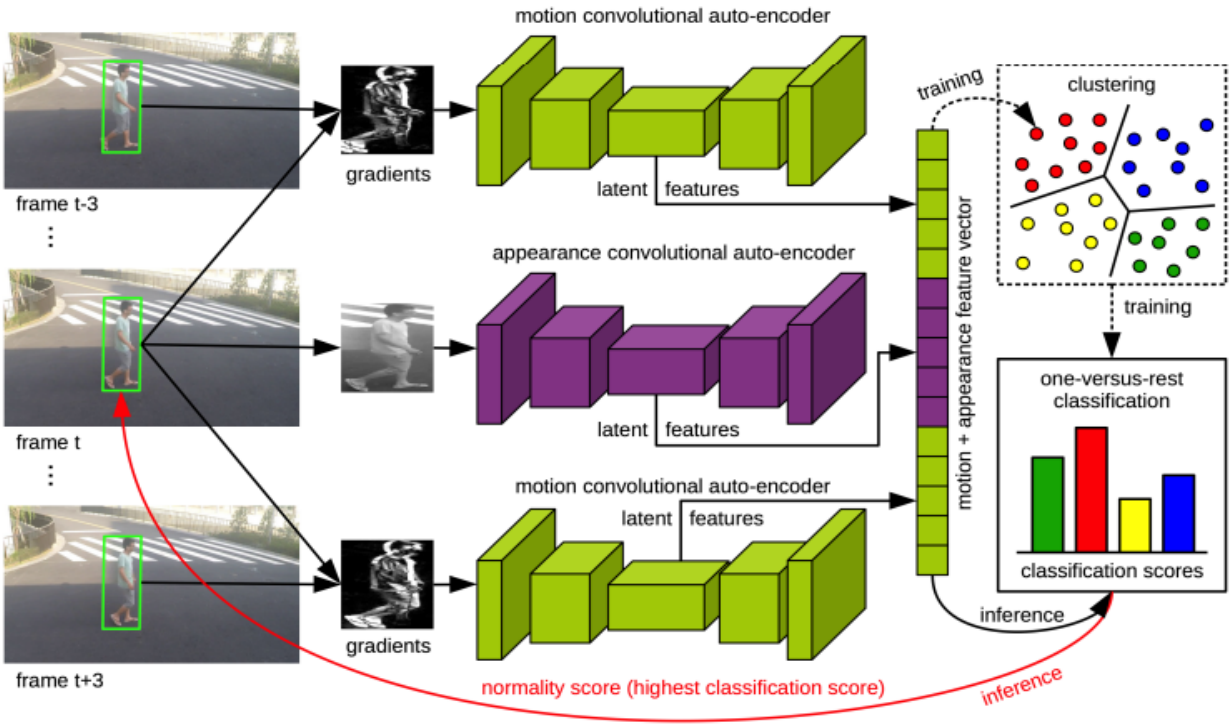


Figure 5: The anomaly detection framework based on training convolutional auto-encoders on top of object detections.

2.3.2.9 Convolutional Transformer based Dual Discriminator Generative Adversarial Networks for Video Anomaly Detection. [11]

This paper presents a methodology called Convolutional Transformer based Dual Discriminator Generative Adversarial Networks (CT-D2GAN) for unsupervised video anomaly detection. The proposed approach addresses the challenges of limited prior knowledge about video anomalies, ineffective capture of normal spatio-temporal patterns, and lack of consideration for local consistency and global coherence in existing methods. The methodology consists of three key components:

Convolutional Transformer: It performs future frame prediction using a convolutional encoder to capture spatial information, a temporal self-attention module to encode temporal dynamics, and a convolutional decoder to integrate spatio-temporal features and predict the future frame. This component aims to capture the normal spatio-temporal patterns effectively and efficiently.

Dual Discriminator Adversarial Training: A dual discriminator-based adversarial training procedure is employed to enhance the future frame prediction. It includes an image discriminator that maintains local consistency at the frame level and a video discriminator that enforces global

coherence of temporal dynamics. This joint training procedure improves the quality and realism of the predicted frames.

Abnormality Identification: The prediction error obtained from the future frame prediction is utilized to identify abnormal video frames. Anomalies are detected by evaluating deviations from the expected normal spatio-temporal patterns.

The effectiveness of the proposed CT-D2GAN framework is demonstrated through thorough empirical studies on three public video anomaly detection datasets: UCSD Ped2, CUHK Avenue, and Shanghai Tech Campus. The results showcase the ability of the methodology to effectively capture normal spatio-temporal patterns and detect anomalies in video sequences.

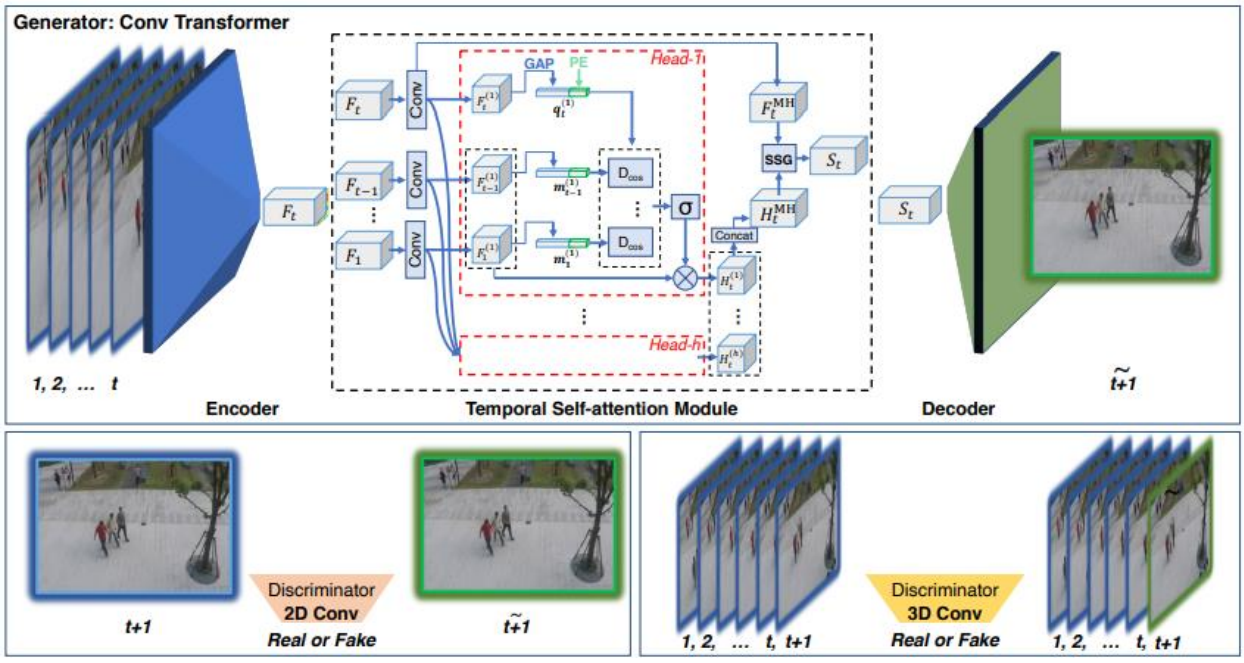


Figure 6: The architecture of the proposed CT-D2GAN framework.

These approaches assume that unseen anomalous videos/images often cannot be reconstructed well and consider samples of high reconstruction errors to be anomalies. However, due to the lack of prior knowledge of abnormality, these approaches can overfit the training data and fail to distinguish abnormal from normal events.

2.4 Weakly-Supervised Learning [12]

2.4.1 Introduction

Weak supervision, also called semi-supervised learning, is a branch of machine learning that combines a small amount of labeled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labeled training data) and supervised learning (with only labeled training data). Semi-supervised learning aims to alleviate the issue of having limited amounts of labeled data available for training.

Semi-supervised learning is motivated by problem settings where unlabeled data is abundant and obtaining labeled data is expensive. Another branch of machine learning that shares the same motivation but follows different assumptions and methodologies is active learning. Unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g., to transcribe an audio segment) or a physical experiment (e.g., determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render large, fully labeled training sets infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

To make any use of unlabeled data, some relationship to the underlying distribution of data must exist. Semi-supervised learning algorithms make use of at least one of the following assumptions:

2.4.1.1 Continuity / smoothness assumption

Points that are close to each other are more likely to share a label. This is also generally assumed in supervised learning and yields a preference for geometrically simple decision boundaries. In the case of semi-supervised learning, the smoothness assumption additionally yields a preference for decision boundaries in low-density regions, so few points are close to each other but in different classes.

2.4.1.2 Cluster assumption

The data tend to form discrete clusters, and points in the same cluster are more likely to share a label (although data that shares a label may spread across multiple clusters). This is a special case of the smoothness assumption and gives rise to feature learning with clustering algorithms.

2.4.1.3 Manifold assumption

Main article: Manifold hypothesis

The data lie approximately on a manifold of much lower dimension than the input space. In this case learning the manifold using both the labeled and unlabeled data can avoid the curse of dimensionality. Then learning can proceed using distances and densities defined on the manifold.

The manifold assumption is practical when high-dimensional data are generated by some process that may be hard to model directly, but which has only a few degrees of freedom. For instance, human voice is controlled by a few vocal folds, and images of various facial expressions are controlled by a few muscles. In these cases, it is better to consider distances and smoothness in the natural space of the generating problem, rather than in the space of all possible acoustic waves or images, respectively.

2.4.2 Weakly Supervised Anomaly Detection

Leveraging some labelled abnormal samples has shown substantially improved performance over the unsupervised approaches.

2.4.2.1 Deep Anomaly Detection with Deviation Networks.[13]

This paper describes a novel anomaly detection framework that addresses limitations in existing deep anomaly detection methods. While deep learning has been successful in various data mining problems, its application to anomaly detection has been relatively limited. Existing methods focus on learning new feature representations, indirectly optimizing anomaly scores, and are typically unsupervised due to the lack of labeled anomaly data. This paper introduces a different approach by proposing an end-to-end learning of anomaly scores through neural deviation learning. The method leverages a few labeled anomalies and a prior probability to enforce statistically significant deviations in anomaly scores compared to normal data objects. This enables the incorporation of prior knowledge when available.

The proposed framework demonstrates improved data efficiency and achieves significantly better anomaly scoring compared to state-of-the-art methods.

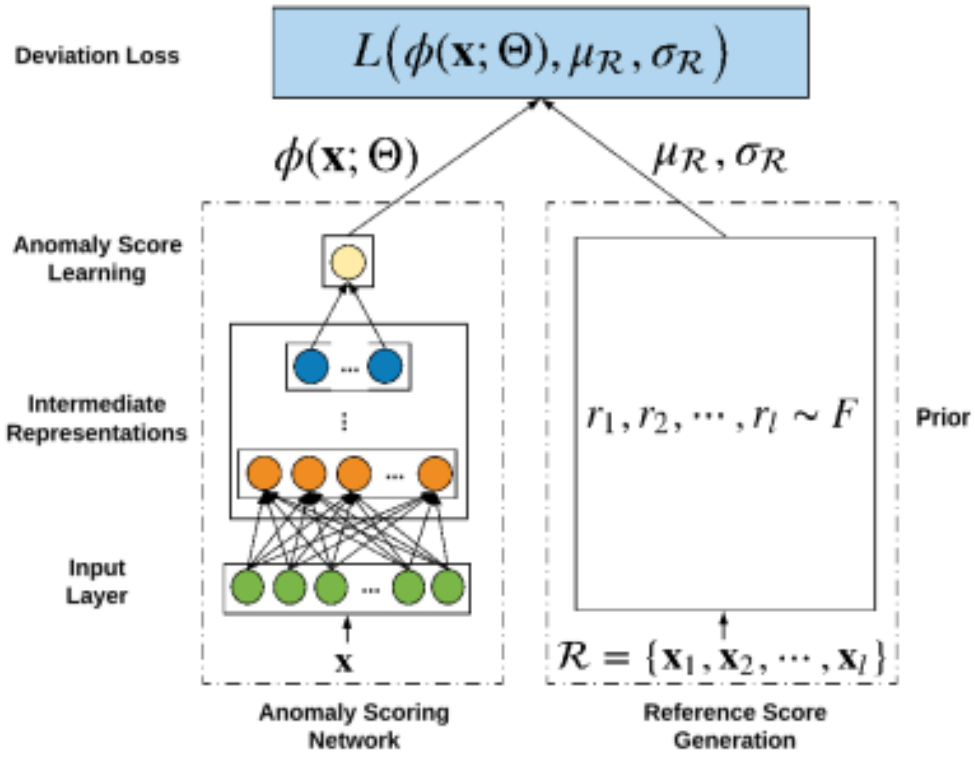


Figure 7: The Proposed Framework.

2.4.2.2 Real-world Anomaly Detection in Surveillance Videos. [14]

This paper proposes a method for learning anomalies in surveillance videos by utilizing both normal and anomalous videos. To avoid the time-consuming process of annotating anomalous segments or clips in training videos, the authors propose a deep multiple instances ranking framework that leverages weakly labeled training videos, where the training labels indicate whether a video is anomalous or normal at the video-level rather than the clip-level. In this approach, normal and anomalous videos are treated as bags, and video segments are considered as instances in multiple instance learning (MIL). The proposed method automatically learns a deep anomaly ranking model that assigns high anomaly scores to anomalous video segments. Additionally, sparsity and temporal smoothness constraints are introduced in the ranking loss function to improve the localization of anomalies during training. By utilizing this framework, the authors aim to learn anomalies effectively without the need for precise annotations at the clip-level.

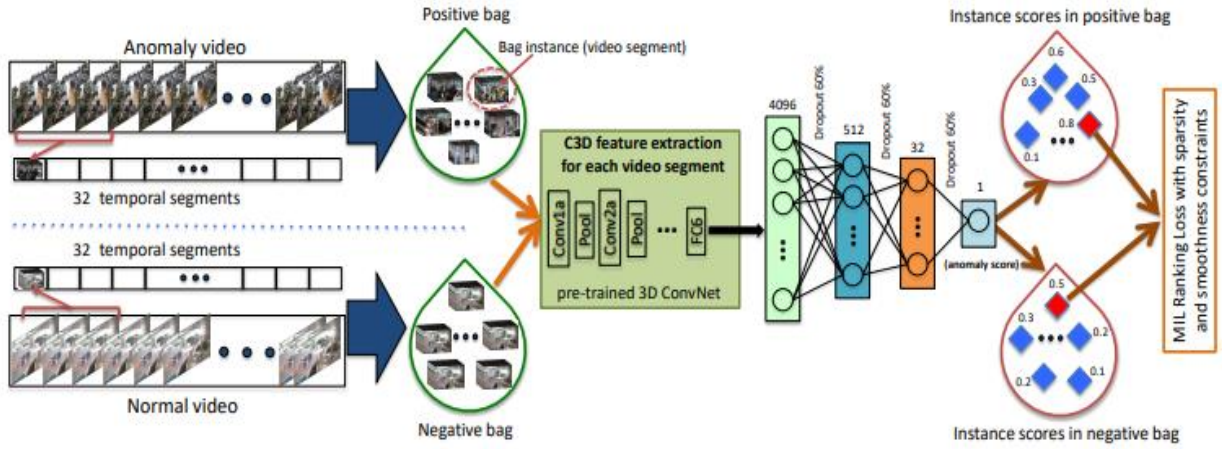


Figure 8: The flow diagram of the proposed anomaly detection approach

2.4.2.3 Few-Shot Anomaly Detection for Polyp Frames from Colonoscopy. [15]

This paper addresses the issue of inappropriate sensitivity to outliers in anomaly detection methods. These methods typically focus on learning the distribution of normal images and classify samples that deviate significantly from this distribution as anomalies. However, they can be sensitive to outliers that are relatively close to the normal images, leading to false detections. To mitigate this, the paper proposes a few-shot anomaly detection method. The method involves training an encoder to maximize mutual information between feature embeddings and normal images, followed by a few-shot score inference network trained with a large set of inliers and a smaller set of outliers. The proposed method is evaluated on the detection of frames containing polyps from colonoscopy video sequences. The training set includes many normal images and a small number of abnormal images. The results demonstrate that the proposed model achieves state-of-the-art detection performance, and the detection performance remains stable after training with approximately 40 abnormal images.

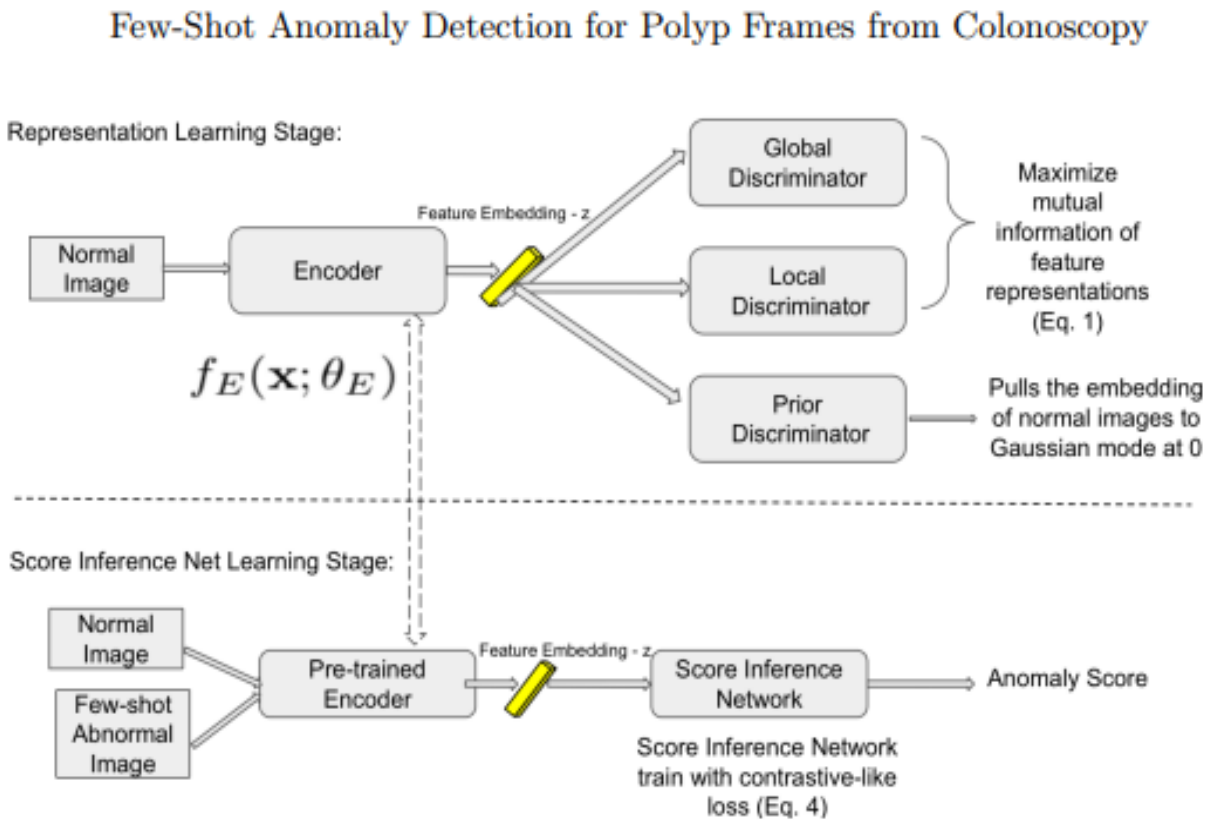


Figure 9: The first stage of FSAD-NET training

2.4.2.4 Not only Look, but also Listen: Learning Multimodal Violence Detection under Weak Supervision. [16]

This paper addresses the limitations of previous work in violence detection in computer vision, which either focus on superficial analysis or lack sufficient data and modalities. To overcome these challenges, the authors first introduce a large-scale and multi-scene dataset called XD-Violence. This dataset comprises 4754 untrimmed videos with audio signals and weak labels, totaling 217 hours of content. The authors then propose a neural network architecture that consists of three parallel branches. The holistic branch captures long-range dependencies using similarity priors, the localized branch captures local positional relations using proximity priors, and the score branch dynamically captures the closeness of predicted scores. Additionally, the method includes an approximator to support online detection. The proposed method outperforms state-of-the-art approaches on the XD-Violence dataset as well as other existing benchmarks. Furthermore, extensive experimental results demonstrate the positive impact of multimodal (audio-visual) input and modeling relationships in violence detection.

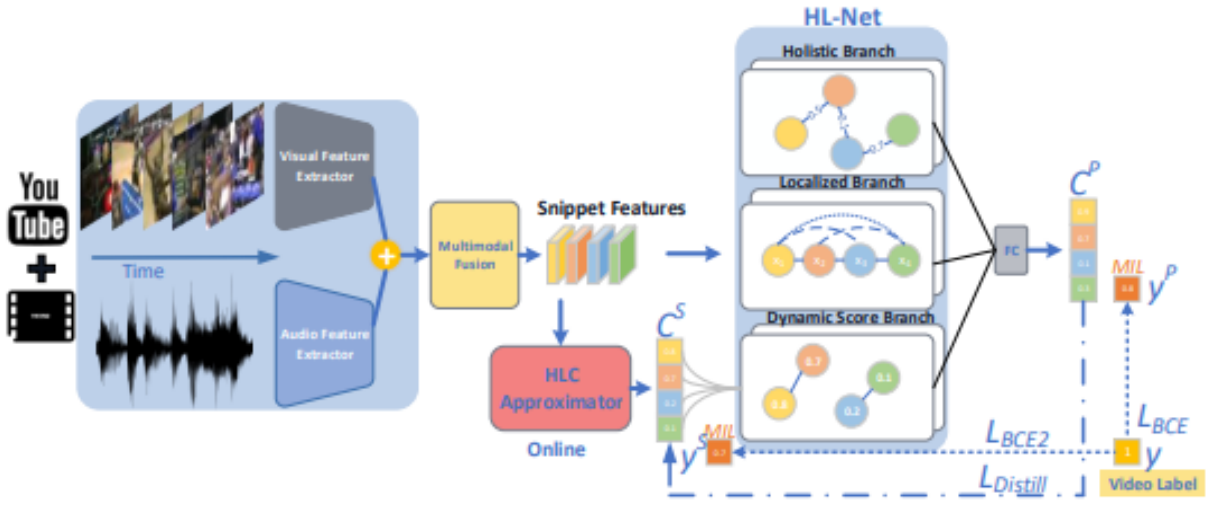


Figure 10: The pipeline of the proposed method.

2.4.2.5 Cleaning Label Noise with Clusters for Minimally Supervised Anomaly Detection. [17]

This paper focuses on the challenging task of detecting anomalous events in real-world videos using only video-level annotations, which often suffer from label noise. To address this issue, the authors propose a weakly supervised anomaly detection method that leverages binary clustering to mitigate the noise in the anomalous video labels. The proposed approach encourages the main network and the clustering to work together, complementing each other in achieving effective weakly supervised training. Experimental results on the UCF-crime and ShanghaiTech datasets demonstrate the superiority of the proposed method, achieving high frame-level AUC scores of 78.27% and 84.16% respectively. This highlights the effectiveness of the method in detecting anomalies using video-level annotations, outperforming existing state-of-the-art algorithms.

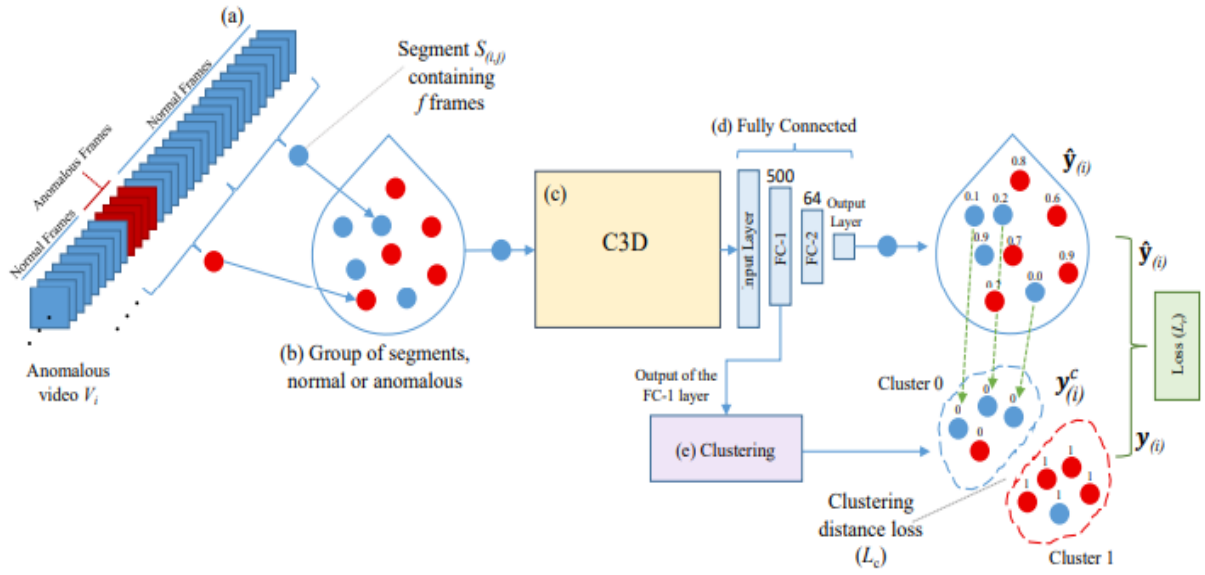


Figure 11: The proposed architecture for anomaly detection in weakly supervised setting.

2.5 Multiple -Instance Learning.[18]

2.5.1 Introduction

In machine learning, multiple-instance learning (MIL) is a type of supervised learning. Instead of receiving a set of instances which are individually labeled, the learner receives a set of labeled *bags*, each containing many instances. In the simple case of multiple-instance binary classification, a bag may be labeled negative if all the instances in it are negative. On the other hand, a bag is labeled positive if there is at least one instance in it which is positive. From a collection of labeled bags, the learner tries to either (i) induce a concept that will label individual instances correctly or (ii) learn how to label bags without inducing the concept.

Babenko (2008)[1] gives a simple example for MIL. Imagine several people, and each of them has a key chain that contains a few keys. Some of these people can enter a certain room, and some aren't. The task is then to predict whether a certain key or a certain key chain can get you into that room. To solve this problem, we need to find the exact key that is common for all the "positive" key chains. If we can correctly identify this key, we can also correctly classify an entire key chain - positive if it contains the required key, or negative if it doesn't.

2.5.1.1 Machine learning

Depending on the type and variation in training data, machine learning can be roughly categorized into three frameworks: supervised learning, unsupervised learning, and reinforcement learning. Multiple instance learning (MIL) falls under the supervised learning framework, where every training instance has a label, either discrete or real valued. MIL deals with problems with incomplete knowledge of labels in training sets. More precisely, in multiple-instance learning, the training set consists of labeled "bags", each of which is a collection of unlabeled instances. A bag is positively labeled if at least one instance in it is positive and is negatively labeled if all instances in it are negative. The goal of the MIL is to predict the labels of new, unseen bags.

2.5.1.2 History

Keeler et al., in his work in the early 1990s was the first one to explore the area of MIL. The actual term multi-instance learning was introduced in the middle of the 1990s, by Dietterich et al. while they were investigating the problem of drug activity prediction.[3] They tried to create a learning system that could predict whether new molecule was qualified to make some drug, or not, through analyzing a collection of known molecules. Molecules can have many alternative low-energy states, but only one, or some of them, is qualified to make a drug. The problem arose because scientists could only determine if molecule is qualified, or not, but they couldn't say exactly which of its low-energy shapes are responsible for that.

One of the proposed ways to solve this problem was to use supervised learning and regard all the low-energy shapes of the qualified molecule as positive training instances, while all the low-energy shapes of unqualified molecules as negative instances. Dietterich et al. showed that such method would have a high false positive noise, from all low-energy shapes that are mislabeled as positive, and thus wasn't really useful.[3] Their approach was to regard each molecule as a labeled bag, and all the alternative low-energy shapes of that molecule as instances in the bag, without individual labels. Thus, formulating multiple-instance learning.

Solution to the multiple instance learning problem that Dietterich et al. proposed is the axis-parallel rectangle (APR) algorithm.[3] It attempts to search for appropriate axis-parallel rectangles constructed by the conjunction of the features. They tested the algorithm on Musk dataset,[4] which is a concrete test data of drug activity prediction and the most popularly used benchmark in multiple-instance learning. The APR algorithm achieved the best result, but APR was designed with Musk data in mind.

The problem of multi-instance learning is not unique to drug finding. In 1998, Maron and Ratan found another application of multiple instances learning to scene classification in machine vision and devised Diverse Density framework. Given an image, an instance is taken to be one or more fixed-size sub images, and the bag of instances is taken to be the entire image. An image is labeled positive if it contains the target scene - a waterfall, for example - and negative otherwise. Multiple instance learning can be used to learn the properties of the sub images which characterize the target scene. From there on, these frameworks have been applied to a wide

spectrum of applications, ranging from image concept learning and text categorization to stock market prediction.[4]

2.5.2 MIL in Anomaly Detection

Multiple Instance Learning (MIL) is a learning framework that has been widely used in anomaly detection. In traditional supervised learning, each training instance is associated with a single label that indicates its class membership. However, in anomaly detection, it is often challenging to obtain accurate labels for individual anomalous instances, as anomalies are typically rare and diverse.

MIL approaches address this challenge by operating at the bag level instead of the instance level. In MIL, a bag is a collection of instances, where each bag is labeled either as normal or anomalous. Importantly, the labels are assigned to bags rather than individual instances. Each bag contains multiple instances, and the label of the bag is determined by the presence or absence of anomalous instances within it.

The key idea in MIL for anomaly detection is that although individual instances within a bag may vary in their characteristics, the overall bag should exhibit anomalous behavior if it contains anomalous instances. This allows MIL algorithms to learn to discriminate between normal and anomalous bags, even without precise labels for individual instances.

During the training phase, MIL algorithms learn to classify bags as normal or anomalous by considering the collective information from the instances within each bag. This allows the model to capture the shared characteristics or patterns that distinguish normal bags from anomalous bags. Various MIL algorithms have been proposed, including instance-level MIL, distance-based MIL, and deep learning-based MIL, each with its own strengths and assumptions.

Overall, multiple instance learning provides a flexible framework for anomaly detection, enabling the learning of anomaly detection models using weak or imprecise labels at the instance level, while leveraging the collective information from bags to make accurate predictions.

2.5.2.1 Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning.[19]

This paper focuses on the problem of anomaly detection using weakly supervised video-level labels, where the goal is to identify abnormal video snippets within a video. This task is typically formulated as a multiple instance learning (MIL) problem, where each video is treated as a bag of snippets. However, existing methods often struggle to effectively recognize positive instances, especially when the abnormal events are subtle and exhibit small differences compared to normal events. This is due to the bias caused by dominant negative instances.

To address this issue, the authors propose a novel method called Robust Temporal Feature Magnitude learning (RTFM). RTFM trains a feature magnitude learning function to improve the recognition of positive instances, making the MIL approach more robust to negative instances from abnormal videos. The method incorporates dilated convolutions and self-attention mechanisms to capture both long- and short-range temporal dependencies, enabling the learning of feature magnitudes in a more accurate manner.

Extensive experiments conducted on four benchmark datasets (ShanghaiTech, UCF-Crime, XD-Violence, and UCSD-Peds) demonstrate the effectiveness of the RTFM-enabled MIL model. It outperforms several state-of-the-art methods by a significant margin, achieving improved subtle anomaly discriminability and sample efficiency. The proposed method addresses the limitations of existing approaches by considering temporal dependencies and enhancing the recognition of positive instances, leading to improved anomaly detection performance.

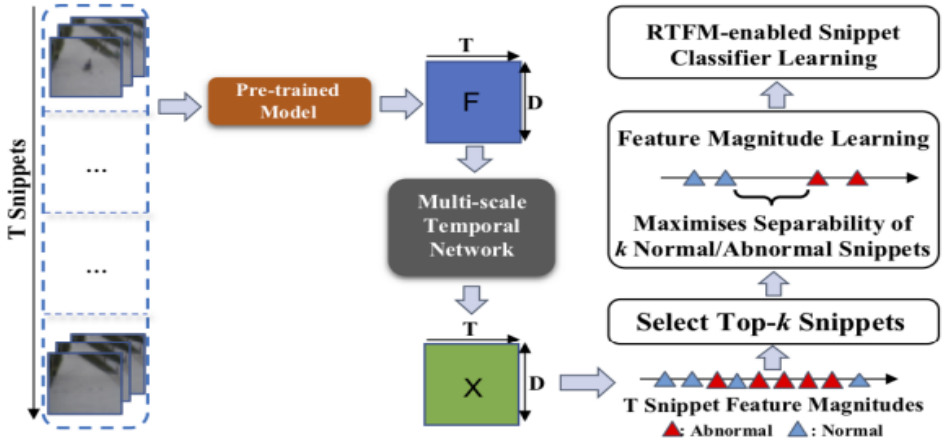


Figure 12: The proposed RTFM architecture.

2.6 Conclusion

In this chapter on related works, several approaches and methods for anomaly detection were discussed. The chapter covered different categories of methods, including supervised learning, unsupervised learning, weakly supervised learning, and multiple-instance learning. Here is a summary of the key findings:

Supervised Learning: Supervised learning approaches typically rely on labeled data to train models for anomaly detection. They use handcrafted features or features extracted from pre-trained deep neural networks to classify instances as normal or abnormal. However, the availability of labeled anomaly data can be limited, and these methods may struggle with detecting subtle anomalies.

Unsupervised Learning: Unsupervised learning methods do not require labeled data and aim to identify anomalies solely based on the characteristics of the data. Traditional unsupervised approaches use handcrafted features and statistical techniques to model normal data distribution and detect deviations. More recent approaches leverage deep learning techniques for better anomaly detection performance.

Weakly-Supervised Learning: Weakly supervised learning methods make use of limited labeled abnormal samples to improve anomaly detection. These approaches leverage prior knowledge or statistical deviations to learn anomaly scores. They offer improved data efficiency and achieve better anomaly scoring compared to unsupervised methods.

Multiple-Instance Learning (MIL): MIL is a learning framework widely used in anomaly detection. MIL approaches operate at the bag level, where a bag represents a collection of instances. The labels are assigned to bags rather than individual instances. This framework allows MIL algorithms to learn discriminative patterns between normal and abnormal bags, even with imprecise or weak instance-level labels. Various MIL algorithms have been proposed, each with its own strengths and assumptions.

In conclusion, the field of anomaly detection encompasses various approaches, each with its own advantages and limitations. Supervised, unsupervised, weakly supervised, and multiple

instance learning methods offer different strategies for detecting anomalies in data. The choice of method depends on the availability of labeled data, the nature of anomalies, and the desired performance. Further research and development in these areas will continue to advance the effectiveness and applicability of anomaly detection techniques.

Chapter 3: System Architecture

3.1 RTFM.[19]

3.1.1 Introduction

Robust Temporal Feature Magnitude learning (RTFM), trains a feature magnitude learning function to effectively recognize the positive instances, substantially improving the robustness of the MIL approach to the negative instances from abnormal videos. RTFM also adapts dilated convolutions and self-attention mechanisms to capture long- and short-range temporal dependencies to learn the feature magnitude more faithfully. Extensive experiments show that the RTFM-enabled MIL model outperforms several state-of-the-art methods by a large margin on four benchmark data sets (ShanghaiTech, UCF-Crime, XD-Violence and UCSD-Peds) and achieves significantly improved subtle anomaly discriminability and sample efficiency.

3.1.2 Method

Robust temporal feature magnitude (RTFM) approach aims to differentiate between abnormal and normal snippets using weakly labelled videos for training. Given a set of weakly-labelled training videos $\mathcal{D} = \{(\mathbf{F}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, where $\mathbf{F} \in \mathcal{F} \subset \mathbb{R}^{T\times D}$ are pre-computed features (e.g., I3D [7] or C3D [61]) of dimension D from the T video snippets, and $y \in \mathcal{Y} = \{0,1\}$ denotes the video-level annotation ($y_i = 0$ if $\mathbf{F}_i$ is a normal video and $y_i = 1$ otherwise). The model used by RTFM is denoted by $r_{\theta,\phi}(\mathbf{F}) = f_\phi(s_\theta(\mathbf{F}))$ and returns a T-dimensional feature $[0,1]^T$ representing the classification of the T video snippets into abnormal or normal, with the parameters θ, ϕ defined below. The training of this model comprises a joint optimisation of an end-to-end multi-scale temporal feature learning, feature magnitude learning and an RTFMenabled MIL classifier training, with the loss.

$$\min_{\theta,\phi} \sum_{i,j=1}^{|\mathcal{D}|} \ell_s\left(s_\theta(\mathbf{F}_i), \left(s_\theta(\mathbf{F}_j)\right), y_i, y_j\right) + \ell_f\left(f_\phi\left(s_\theta(\mathbf{F}_i)\right), y_i\right),$$

where sθ : F → X is the temporal feature extractor (with X ⊂ R T ×D), fφ : X → [0, 1]T is the snippet classifier, `s(.) denotes a loss function that maximizes the separability between the top-k snippet features from normal and abnormal videos, and `f (.) is a loss function to train the snippet classifier fφ(.) also using the top-k snippet features from
normal and abnormal videos. Next, we discuss the theoretical motivation for our proposed RTFM, followed by a detailed description of the approach.

3.1.3 Theoretical Motivation

Top-k MIL in [25] extends MIL to an environment where positive bags contain a minimum number of positive samples and negative bags also contain positive samples, but to a lesser extent, and it assumes that a classifier can separate positive and negative samples. Our problem is different because negative bags do not contain positive samples, and we do not make the classification separability assumption. Following the nomenclature introduced above, a temporal feature extracted from a video is denoted by X = sθ(F) in (1), where snippet features are represented by the rows xt of X. An abnormal snippet is denoted by x + ∼ P + x (x), and a normal snippet, x − ∼ P − x (x). An abnormal video X+ contains μ snippets drawn from P + x (x) and (T − μ) drawn from P − x (x), and a normal video X− has all T snippets sampled from P − x (x). To learn a function that can classify videos and snippets as normal or abnormal, we define a function that classifies a snippet using its magnitude (i.e., we use `2 norm to compute the feature magnitude), where instead of assuming classification separability between normal and abnormal snippets (as assumed in [25]), we make a milder assumption that E[kx +k2] ≥ E[kx −k2]. This means that by learning the snippet feature from sθ(F), such that normal ones have smaller feature magnitude than abnormal ones, we can satisfy this assumption. To enable such learning, we rely on an optimisation based on the mean feature magnitude of the top k snippets from a video [25], defined by

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

where $g_{\theta,k}(.) is\ parameterised\ by\ \theta$ to indicate its dependency on $s_\theta(.) to produce \mathbf{x}_t, \Omega_k(\mathbf{X})$ contains a subset of k snippets from $\{\mathbf{x}_t\}_{t=1}^{T}$ and $|\Omega_k(\mathbf{X})| = k$. The separability between abnormal and normal videos is denoted by

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

For the theorem below, we define the probability that a snippet from $\Omega_k(\mathbf{X}^+)$is abnormal with $p_k^+(\mathbf{X}^+) = \frac{\min(\mu,k)}{k+\epsilon}$, with $\epsilon > 0$ and from normal $\Omega_k(\mathbf{X}^-), p_k^+(\mathbf{X}^-) = 0$. This definition means that it is likely to find an abnormal snippet within the top k snippets in $\Omega_k(\mathbf{X}^+)$, as long as $k \leq \mu$.

Theorem : (Expected Separability Between Abnormal and Normal Videos). Assuming that $\mathbb{E}[\|\mathbf{x}^+\|_2] \geq \mathbb{E}[\|\mathbf{x}^-\|_2]$, where $\mathbf{X}^+$has μ abnormal samples and $(T - \mu)$ normal samples, where $\mu \in [1, T]$, and $\mathbf{X}^-$has T normal samples. Let
$D_{\theta,k}(.)$be the random variable from which the separability scores $d_{\theta,k}(.) of (3) are drawn$[25].

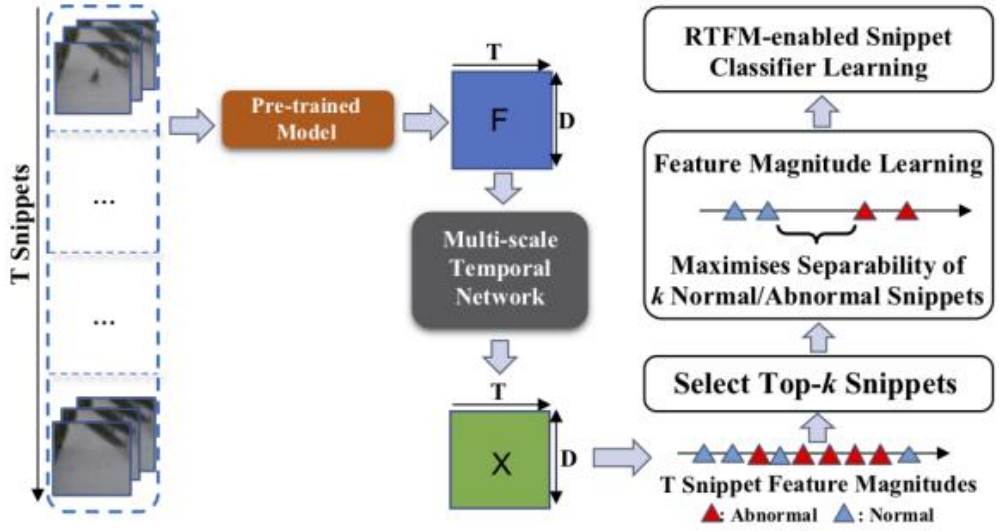


Figure 13: The proposed RTFM architecture.

3.2 I3D.[20]

Two-Stream Inflated 3D ConvNet (I3D) is based on 2D ConvNet inflation: filters and pooling kernels of very deep image classification ConvNets are expanded into 3D, making it possible to learn seamless spatio-temporal feature extractors from video while leveraging successful ImageNet architecture designs and even their parameters. It is shown that, after pre-training on Kinetics dataset, I3D models considerably improve upon the state-of-the-art in action classification, reaching 80.9% on HMDB-51 and 98.0% on UCF-101.

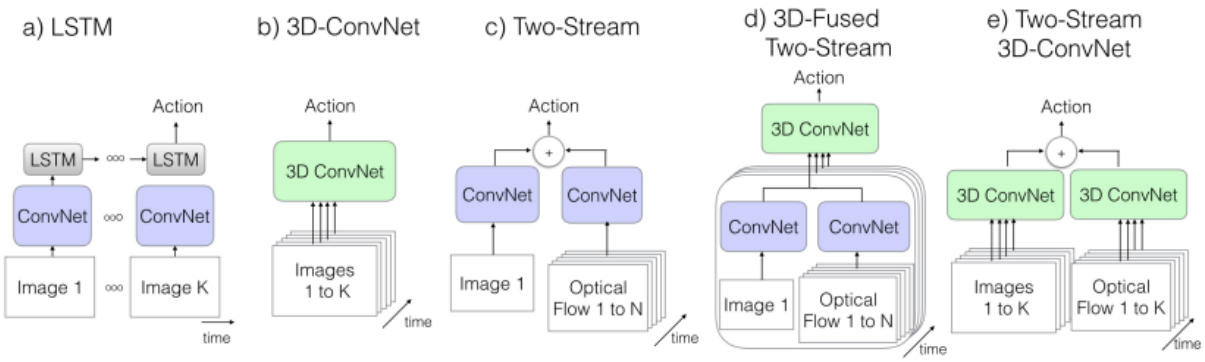


Figure 14: The Video architectures considered in this paper.

3.3 Architecture

3.3.1 introduction

This chapter presents a comprehensive overview of the architecture underlying the surveillance system, highlighting the key components and technologies employed for efficient and effective video analysis. The system architecture encompasses three main stages: anomaly prediction, object detection, and feature extraction.

The anomaly prediction stage utilizes the Robust Temporal Feature Magnitude (RTFM) model to compute anomaly scores for each frame of the video footage. The RTFM model leverages deep learning techniques, incorporating dilated convolutions and self-attention mechanisms, to capture both long- and short-range temporal dependencies. By learning feature magnitudes accurately, the RTFM model enhances the recognition of positive instances and improves the discrimination of subtle anomalies. The anomaly scores provide a quantitative measure of the likelihood that a displayed frame contains an anomaly, enabling timely identification of abnormal events.

In the object detection stage, the system employs the Ultralytics YOLOv8 model, a state-of-the-art object detection algorithm. By analyzing the video frames, YOLOv8 identifies and localizes individuals within the scene by drawing bounding boxes around them. The model is trained on diverse datasets and is customizable based on the specific requirements of the surveillance task. Different weights can be selected by the observer to balance the trade-off between detection accuracy and computational efficiency, allowing flexibility in adapting to various surveillance scenarios.

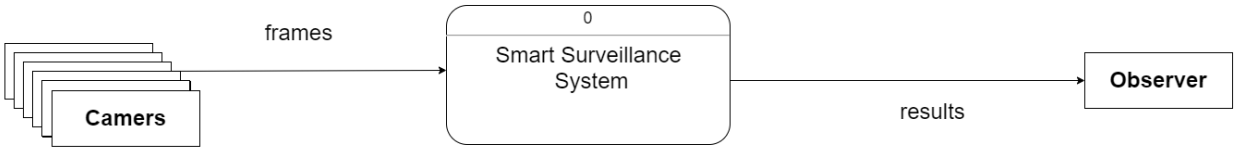
The final stage of the system architecture involves feature extraction using the I3D (Inflated 3D) model. I3D is a powerful deep neural network that leverages spatiotemporal information to capture motion dynamics in video data. By extracting high-level features from the video frames, the system gains a rich representation of the scene, which can be further utilized for tasks such as activity recognition, abnormal behavior detection, and scene understanding. The extracted features provide valuable contextual information that aids in the accurate interpretation of the surveillance environment.

By integrating these components and technologies, the system architecture enables comprehensive video analysis, empowering observers with real-time insights and actionable information. The fusion of anomaly prediction, object detection, and feature extraction facilitates a holistic understanding of the surveillance scene, facilitating effective decision-making and proactive response to potential security threats. Throughout this chapter, we will delve into the detailed implementation and interaction of these components, elucidating the system's architecture and its significance in ensuring robust surveillance and monitoring capabilities.

3.3.2 Data Flow Diagram

3.3.2.1 Context Level

The system operates by utilizing two external sources: cameras and an observer. The cameras capture frames, which are then input into the system. These frames are processed within the system, which subsequently generates results. These results are then communicated back to the observer for further analysis or action.



3.3.2.2 Level 0

At this level, we delve into the internal components of the system to understand the flow of data and operations. As depicted in Figure [x], the camera frames (Process 1.0) serve as the primary input to the system. These frames are then simultaneously received by two key processes within the system.

Process 2.0, responsible for anomaly detection, analyzes the frames and generates an anomaly score that represents the likelihood of each frame being considered an anomaly. The anomaly scores serve as one of the outputs of Process 2.0.

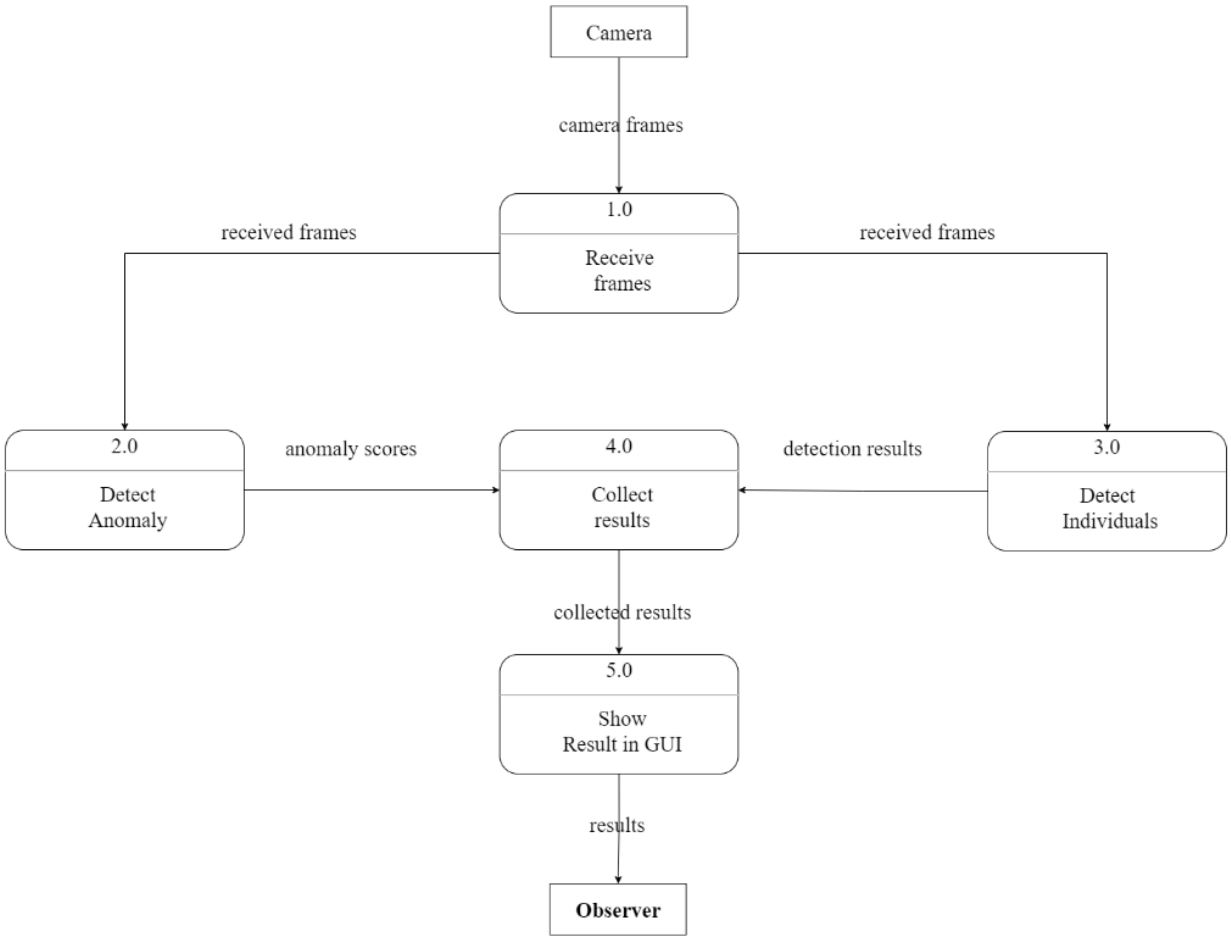
Simultaneously, Process 3.0 focuses on detecting individuals within the frames. By utilizing advanced algorithms, such as the Ultralytics YOLOv8 model, it identifies individuals and provides detection results, including bounding boxes or other relevant information. The detection results constitute the second output of Process 3.0.

To consolidate the outputs from Processes 2.0 and 3.0, the system employs Process 4.0, which merges the anomaly scores and detection results. By combining these pieces of information, the system enhances the overall understanding of the observed frames.

Subsequently, the merged outputs are directed to Process 5.0, which is responsible for displaying the results in the graphical user interface (GUI). The GUI serves as the interface between the system and the observer, presenting the consolidated information in a visually accessible manner.

The displayed results in the GUI are then communicated back to the observer, enabling real-time monitoring and analysis. This feedback loop ensures that the observer has access to the processed information, allowing them to make informed decisions or take appropriate actions based on the system's outputs.

3.3.2.3 Level 1: RTFM



In this Level, we focus on the detailed process of detecting anomalies and computing anomaly scores. The frames received from Process 1.0 undergo several preprocessing steps before being analyzed.

Process 2.1 is responsible for preprocessing the frames. It applies various techniques to enhance the quality and prepare the frames for further analysis.

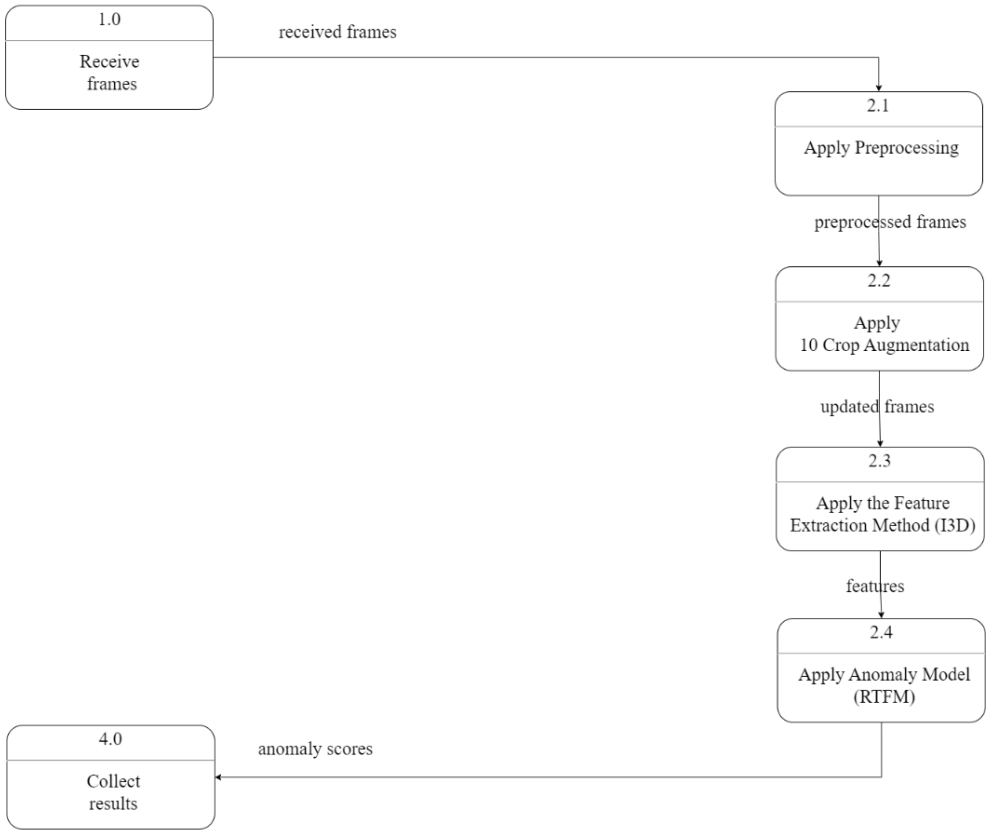
Once preprocessed, the frames are passed to Process 2.2, which applies a 10 crop augmentation technique. This augmentation method involves creating multiple variations of each frame by taking ten different crops, resulting in an increased diversity of data for improved anomaly detection accuracy.

The augmented frames are then forwarded to Process 2.3, where the I3D model is utilized. The I3D model extracts essential features from the frames, capturing temporal information and spatial details that are crucial for anomaly detection.

Subsequently, the extracted features are fed into Process 2.4, which employs the RTFM model. This model utilizes the extracted features to compute anomaly scores for each frame. The anomaly scores quantify the likelihood of a frame being classified as an anomaly based on its unique characteristics and patterns.

The computed anomaly scores are then collected in Process 4.0, where they are stored or further processed for subsequent analysis or visualization. Process 4.0 serves as a central point for aggregating the anomaly scores from the previous steps, facilitating efficient management and utilization of the anomaly detection results.

By examining this Level 1 DFD diagram, we gain a more detailed understanding of the internal processes involved in detecting anomalies and computing anomaly scores. The diagram highlights the specific steps of frame preprocessing, augmentation, feature extraction, anomaly score computation, and result collection.



3.3.2.4 Level 1: YOLO

In this Level 1 DFD diagram, we focus on the details of detecting individuals and extracting their boundary box information using the YOLO detection model. The frames received from Process 1.0 are passed to Process 3.1 for analysis.

Process 3.1 utilizes the YOLO detection model to identify and locate individuals within the frames. The model processes the frames and outputs boundary box information, specifying the coordinates and dimensions of each detected individual.

The boundary box information is then simultaneously fed into two processes: Process 3.2 and Process 3.3.

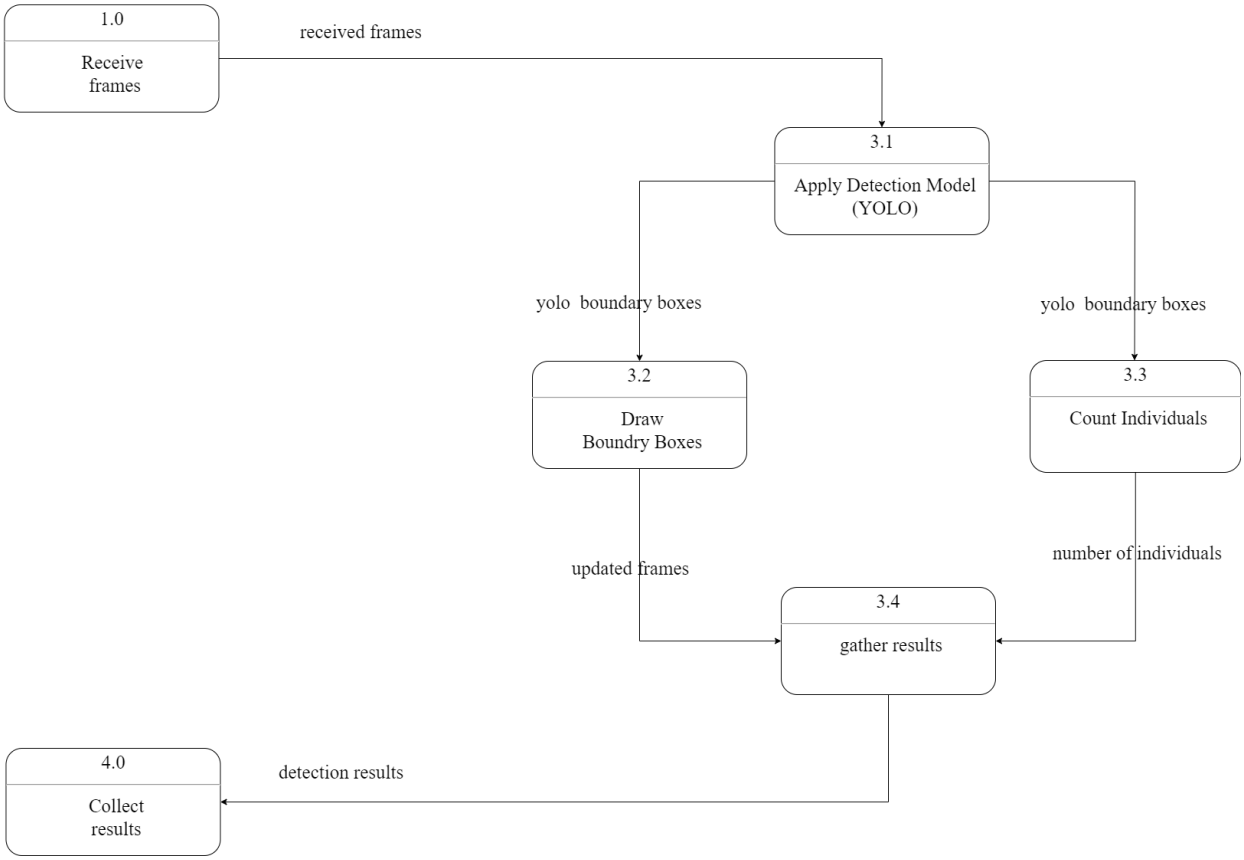
Process 3.2 is responsible for drawing the boundary boxes on the frames. This process overlays the boundary boxes onto the corresponding frames, visually highlighting the detected individuals for better visualization and understanding.

Meanwhile, Process 3.3 counts the number of individuals represented by the boundary boxes. It analyzes the boundary box information and determines the total count of individuals detected within the frames.

Both the frames with drawn boundary boxes from Process 3.2 and the count of individuals from Process 3.3 are then forwarded to Process 3.4. Process 3.4 serves as a central hub, gathering the results from the previous steps.

Finally, the gathered results from Process 3.4 are sent to Process 4.0. Process 4.0 is responsible for collecting and further processing the results for subsequent actions, such as displaying the collected information in the GUI or communicating it to other parts of the system.

By examining this Level 1 DFD diagram, we gain a more detailed understanding of the internal processes involved in detecting individuals, extracting boundary box information, drawing the boxes on frames, counting the number of individuals, and collecting the results. The diagram showcases the flow of data and interactions between the various processes in this specific level of the system architecture.



3.3.3 Conclusion

The system architecture presented in Chapter 3 provides a comprehensive framework for efficient and effective video analysis in the context of surveillance. The architecture is designed to address the challenges of anomaly prediction, object detection, and feature extraction, enabling robust surveillance and monitoring capabilities.

At the anomaly prediction stage, the Robust Temporal Feature Magnitude (RTFM) model is employed to compute anomaly scores for each frame of the video footage. By leveraging deep learning techniques, including dilated convolutions and self-attention mechanisms, the RTFM model captures both long- and short-range temporal dependencies, enhancing the recognition of positive instances and improving the discrimination of subtle anomalies. The anomaly scores serve as a quantitative measure of the likelihood of anomalies, facilitating timely identification of abnormal events.

In the object detection stage, the system utilizes the Ultralytics YOLOv8 model, a state-of-the-art object detection algorithm. By analyzing the video frames, YOLOv8 identifies and localizes individuals within the scene by drawing bounding boxes around them. The model's flexibility allows customization based on specific surveillance requirements, striking a balance between detection accuracy and computational efficiency.

The final stage of the system architecture involves feature extraction using the I3D (Inflated 3D) model. I3D leverages spatiotemporal information to capture motion dynamics in video data, extracting high-level features that provide valuable contextual information for tasks such as activity recognition, abnormal behavior detection, and scene understanding. These features enhance the accurate interpretation of the surveillance environment.

The integration of anomaly prediction, object detection, and feature extraction components empowers observers with real-time insights and actionable information. The fusion of anomaly scores, detection results, and extracted features facilitates a holistic understanding of the surveillance scene, enabling effective decision-making and proactive response to potential security threats.

Throughout this chapter, we have examined the detailed implementation and interaction of the system's components, elucidating the significance of each process in ensuring robust surveillance and monitoring capabilities. By following the data flow and operations at different levels, we have gained a comprehensive understanding of the system architecture and its role in comprehensive video analysis.

The presented system architecture has demonstrated its effectiveness through extensive experiments, outperforming several state-of-the-art methods on benchmark datasets. It has showcased improved subtle anomaly discriminability, sample efficiency, and the ability to handle diverse surveillance scenarios.

In conclusion, the system architecture outlined in Chapter 3 provides a powerful and reliable framework for surveillance systems. By leveraging advanced techniques in anomaly prediction, object detection, and feature extraction, the architecture enhances the understanding and interpretation of video data, empowering observers with real-time insights for effective decision-making and proactive response to potential security threats.

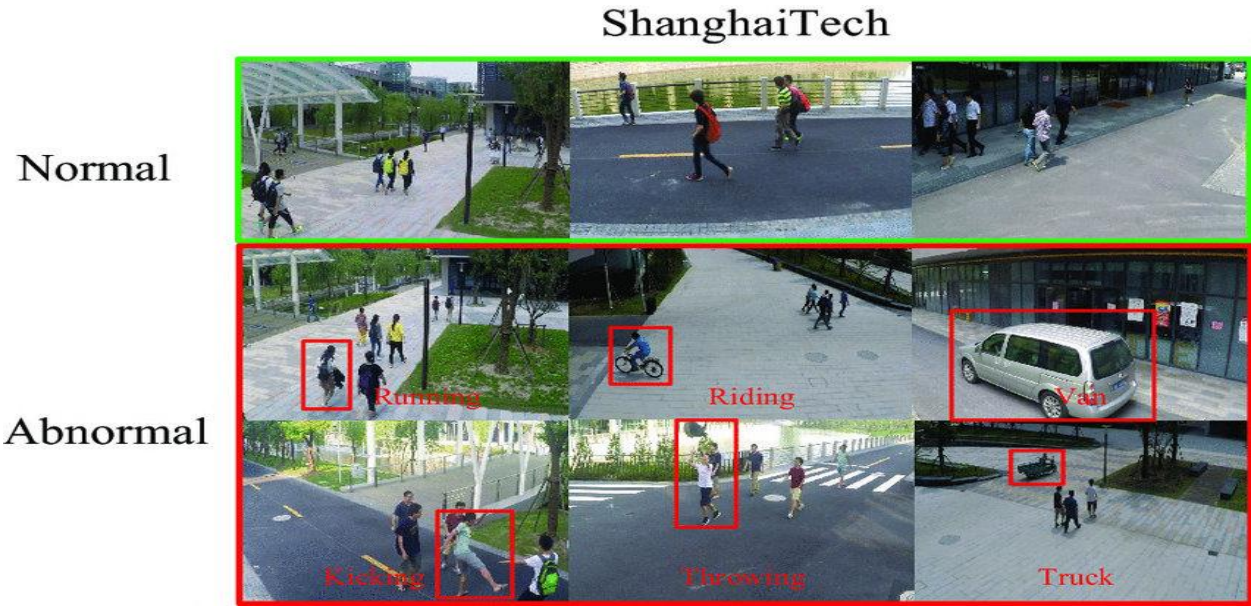
Chapter 4: System Implementation and Results

4.1 Data Set Description: ShanghaiTech

The ShanghaiTech data set is a medium-scale collection of fixed-angle street video surveillance footage. It encompasses a total of 437 videos, with 13 distinct background scenes. Among these videos, 307 are classified as normal, while the remaining 130 videos contain anomalies. The original data set has gained significant recognition as a benchmark for anomaly detection tasks, assuming the availability of normal training data. However, Zhong et al. undertook a reorganization of the data set, wherein a subset of anomalous testing videos was selected and incorporated into the training data. This restructuring facilitated the creation of a weakly supervised training set, ensuring that both the training and testing sets covered all 13 background scenes. The conversion process to adapt ShanghaiTech for the weakly supervised setting precisely follows the procedure outlined in the work by Zhong et al.

Please note that additional details about the data set, such as its source, specific characteristics, and any other relevant information, can be included in this section.

Fig x below shows the sample normal and abnormal clips from the dataset.



4.2 Description of Software Tools Used

4.2.1 Feature Extraction

- **Tool:** i3d
- **Purpose:** The i3d tool is utilized for feature extraction from video surveillance data. It helps in capturing spatio-temporal features from the input videos, enabling subsequent analysis and anomaly detection.

4.2.2 Augmentation

- **Tool:** 10 crop augmentation
- **Purpose:** The 10-crop augmentation technique is employed to augment the input data set. It involves extracting ten different crops from each image to enhance the diversity of the training data, leading to improved model performance and generalization.

4.2.3 Anomaly Detection

- **Tool:** MIL Approach with the RTFM Model
- **Purpose:** The MIL (Multiple Instance Learning) approach, implemented using the RTFM (Real-Time Foreground-Background Modeling) model, is utilized for anomaly detection. This approach allows the detection of anomalies by modeling the foreground and background information in the video surveillance data, enabling accurate identification of unusual events or behaviors.

4.2.4 PyQt5

- **Purpose:** PyQt5 is a Python library used for creating desktop applications with a graphical user interface (GUI). It provides the necessary components and widgets for building the user interface of the surveillance system, facilitating user interaction and control.

4.2.5 pyshine

- **Purpose:** pyshine is a Python library that offers various utilities and functions for visual enhancements in GUI applications. It is utilized in this project to display text overlays and annotations on the video frames, such as the number of people detected and the anomaly state.

4.2.6 ultralytics

- **Purpose:** ultralytics is a Python library specifically designed for object detection tasks. It provides convenient functions and interfaces for utilizing pre-trained object detection models, such as YOLO (You Only Look Once), which is employed in this project to detect objects in the video frames.

4.2.7 timm

- **Purpose:** timm is a PyTorch library that offers a wide range of pre-trained models for computer vision tasks. In this project, timm is likely used to access pre-trained models for feature extraction or other relevant tasks.

4.2.8 einops

- **Purpose:** einops is a Python library that provides flexible and efficient operations for reshaping, reordering, and combining tensors. It is likely used in this project for data preprocessing or manipulation tasks.

4.2.9 ftfy

- **Purpose:** ftfy is a Python library used for handling and fixing Unicode text-related issues. It is employed in this project to ensure proper handling of text data and to address any encoding or formatting problems that may arise.

4.2.10 mmcv

- **Purpose:** mmcv is an open-source computer vision library that offers a comprehensive suite of functions and utilities for various computer vision tasks. It is likely used in this project for image and video processing operations, such as reading video frames or performing transformations.

4.2.11 pyyaml

- **Purpose:** pyyaml is a YAML parser and emitter library for Python. It allows for easy parsing and handling of YAML files, which may be used in this project for configuration purposes or storing certain parameters.

4.2.12 tqdm

- **Purpose:** tqdm is a Python library that provides a progress bar utility for tracking the progress of iterative processes. It is likely used in this project to display progress information during tasks such as data loading, training, or inference.

4.2.13 munch

- **Purpose:** munch is a Python library that provides a dictionary-like interface with attribute-style access. It allows for more convenient handling and accessing of data structures, which may be used in this project for managing and processing various configurations or settings.

4.2.14 terminaltables

- **Purpose:** terminaltables is a Python library that assists in generating ASCII tables for terminal-based applications. It might be used in this project to display tabular information or results in the command-line interface.

4.2.15 scikit-learn

- **Purpose:** scikit-learn is a popular machine learning library in Python. It provides a wide range of machine learning algorithms and utilities, which can be utilized in this project for tasks such as data preprocessing, model evaluation, or performance metrics calculation.

4.2.16 pandas

- **Purpose:** pandas are a powerful data manipulation and analysis library in Python. It offers various data structures and functions for handling structured data, such as data frames, which may be used in this project for data preprocessing, analysis, or result representation.

4.2.17 termcolor

- **Purpose:** termcolor is a Python library that enables the printing of colored text and formatting in the terminal. It might be used in this project to add color-coded text or highlights for specific information or notifications.

4.2.18 typed-argument-parser

- **Purpose:** typed-argument-parser is a Python library that provides an easy-to-use interface for parsing command-line arguments with type annotations. It simplifies the process of handling command-line arguments in the project, allowing for clearer argument definition and validation.

4.3 The hardware used for the project

4.3.1 Graphics Card

- **Model:** GTX 1660
- **Purpose:** The GTX 1660 is a graphics card designed for gaming and general-purpose computing tasks. It offers good performance and is capable of handling demanding graphics-related operations, such as image and video processing, which are essential for tasks like object detection and anomaly detection in video surveillance.

4.3.2 Processor (CPU)

- **Model:** Intel Core i8 (Assuming you meant Intel Core i7)
- **Purpose:** The Intel Core i7 processor is a high-performance CPU known for its multi-threading capabilities and overall processing power. It provides fast and efficient computation, allowing for smooth execution of complex algorithms and tasks involved in the video surveillance system.

4.3.3 Random Access Memory (RAM)

- **Size:** 16 GB (Assuming you meant 16 GB of RAM)
- **Purpose:** RAM is essential for storing and accessing data during program execution. With 16 GB of RAM, the system has enough memory to accommodate the data processing requirements of the video surveillance system. This enables smooth and efficient data manipulation, model training, and inference.

4.4 The experimental setup and results obtained from the project

- **The learning rate of 0.001 was used for updating the model parameters during training. The learning rate controls the step size at each iteration and influences the speed and quality of the convergence of the model.**

The experimental results indicate that the proposed RTFM (Real-Time Feature Magnitude) method achieved superior performance compared to state-of-the-art unsupervised learning methods and weakly supervised approaches. The results are summarized as follows:

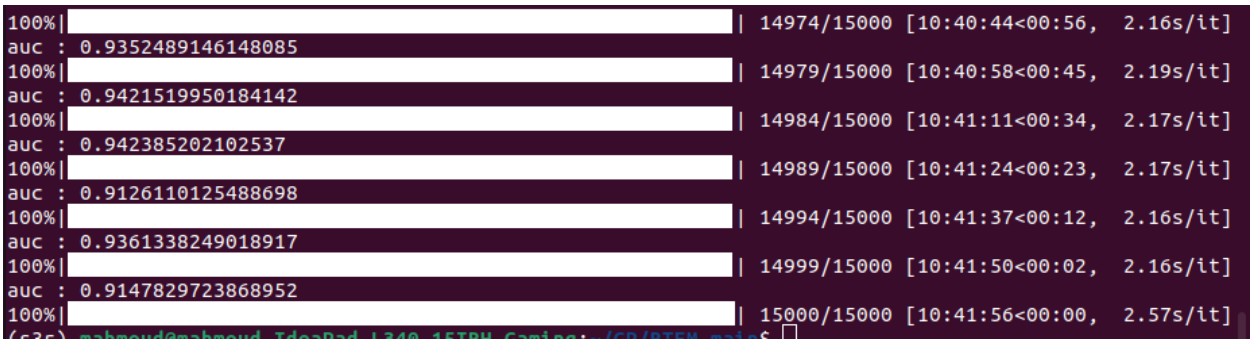
- **Using I3D-RGB features, the RTFM model achieved the best Area Under the Curve (AUC) result on the dataset with a score of 97.21%. This demonstrates the effectiveness of the RTFM method in leveraging I3D-RGB features for anomaly detection.**

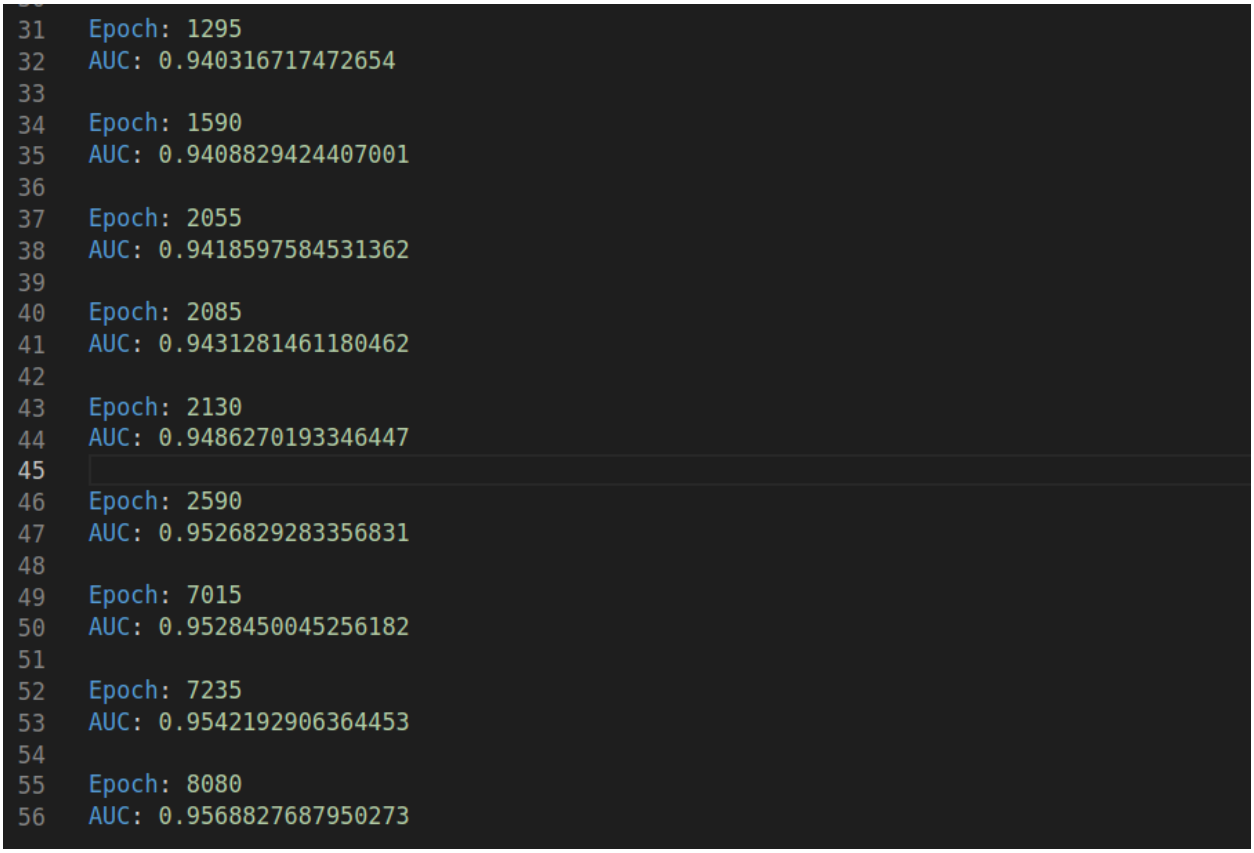
- **The RTFM-enabled Multiple Instance Learning (MIL) method outperformed current state-of-the-art MIL-based methods by 10% to 14% when using the same I3D-RGB features. Even when compared to methods relying on more advanced feature extractors like I3D-RGB and I3D-Flow, the RTFM model still achieved more than a 5% improvement in performance. These results highlight the benefits of the proposed feature magnitude learning.**

- **The RTFM method also outperformed a Graph Convolutional Network (GCN)-based weakly supervised method by 11.7%. This suggests that the proposed MTN (Magnified Temporal Neglect) module in the RTFM model is more effective at capturing temporal dependencies than GCN.**

- **Additionally, when considering C3D-RGB features, the RTFM model achieved the state-of-the-art AUC score of 91.51%. This significant improvement surpasses the performance of previous methods utilizing C3D-RGB features.**

These results demonstrate the effectiveness of the RTFM method in anomaly detection, outperforming existing approaches and achieving state-of-the-art performance on the given dataset.

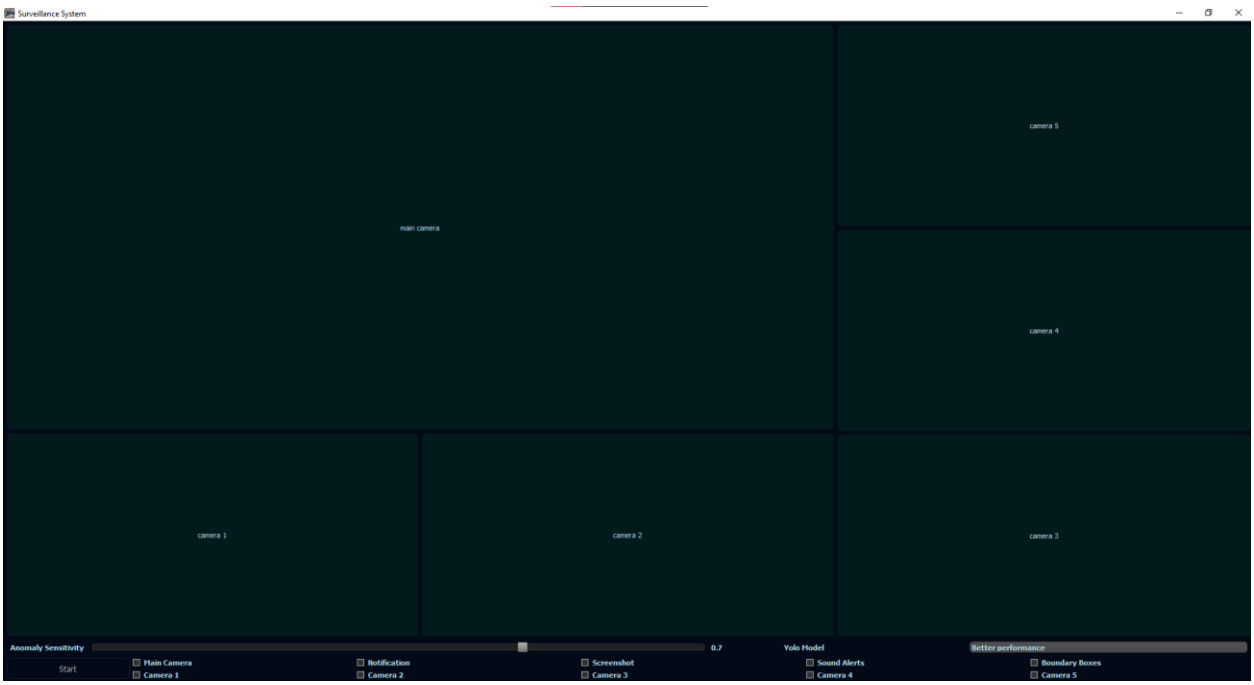




Chapter 5: Visual Walkthrough

5.1 Initial state

The initial state of the anomaly surveillance system displays a user interface with the following components on the screen: five cameras, a toolbar featuring various options related to camera operations, and a slider for adjusting the anomaly score. The five cameras are visually presented, providing real-time feeds from their respective locations. The toolbar offers a range of functionalities, allowing users to interact with the cameras effectively. Additionally, the presence of a slider enables users to fine-tune the anomaly score, which influences the system's sensitivity in detecting unusual events or behaviors.



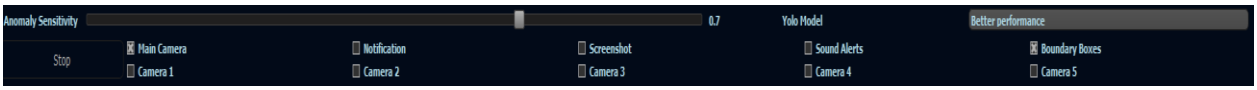
5.2 Toolbar

The toolbar in the anomaly surveillance system offers various options for enhanced control and functionality. Users can individually enable or disable each camera, tailoring the system's monitoring capabilities based on specific requirements. Furthermore, users can opt to draw bounding boxes around everyone detected by the system, aiding in visual tracking and identification.

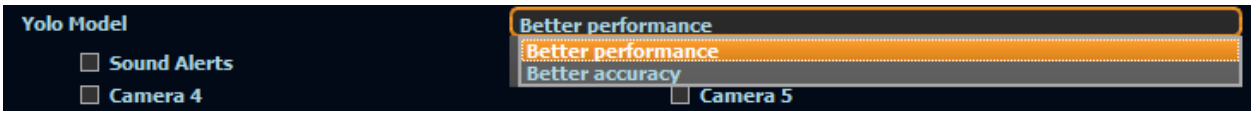
In addition, the toolbar provides several actions that can be triggered when an anomaly is detected. Users can choose to take a screenshot of the anomaly, capturing crucial evidence for later analysis. The option to send an email notification is available, enabling immediate alerts to be sent to designated recipients when an anomaly occurs. Furthermore, an alert sound can be played, providing an audible warning signal to draw attention to the anomaly event.

To further customize the anomaly detection process, the toolbar includes an anomaly sensitivity slider. Users can adjust this slider to set a threshold value for the anomaly score. If the anomaly score exceeds or equals the set threshold, the system will activate all the enabled actions mentioned above.

Lastly, the toolbar features a convenient start/stop button, allowing users to initiate or halt the program's execution with ease, ensuring efficient operation and control of the anomaly surveillance system.



The anomaly surveillance system provides users with the flexibility to choose their preferred YOLO (You Only Look Once) model, based on specific requirements. Users can select between a model that prioritizes better performance or one that emphasizes better accuracy. This allows users to optimize the system's object detection capabilities according to their specific needs, ensuring optimal results in anomaly detection and monitoring.



5.3 Normal State

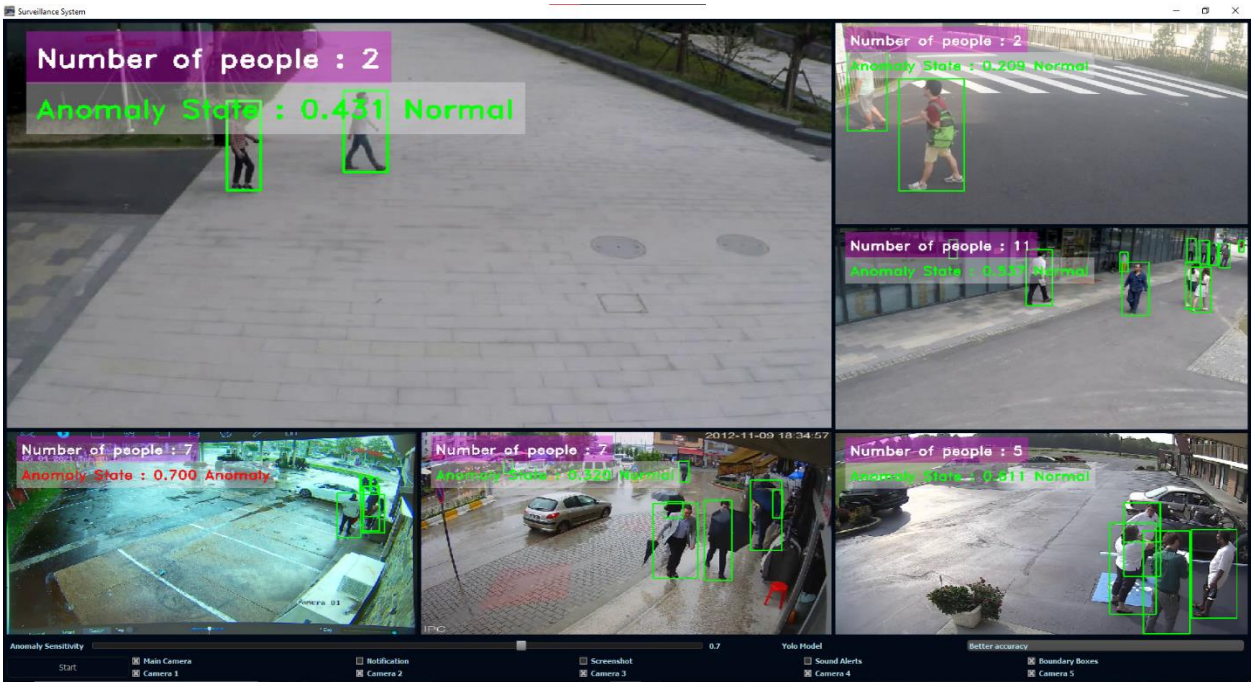
When a camera is enabled within the anomaly surveillance system and bounding boxes are activated, the system's display showcases vital information. This includes the number of individuals detected within the camera's field of view, the anomaly score associated with the current scene, and the bounding boxes outlining the identified objects or persons of interest. These visual indicators aid in monitoring and analyzing the scene, providing valuable insights into potential anomalies or abnormal activities within the camera's vicinity. The combination of real-time video feed, individual count, anomaly score, and bounding boxes enhances the system's capability to detect and alert users about potential abnormalities effectively.

5.4 Anomaly State

When an anomaly occurs in the anomaly surveillance system, a distinct anomaly state is triggered. As part of the visual feedback, the text color within the system undergoes a noticeable change. This color alteration serves as a prominent indicator, drawing immediate attention to the occurrence of an anomaly. By dynamically adjusting the text color, the system effectively communicates the presence of abnormal events or behaviors, allowing users to promptly identify and respond to potential security concerns or unusual activities within the monitored environment.

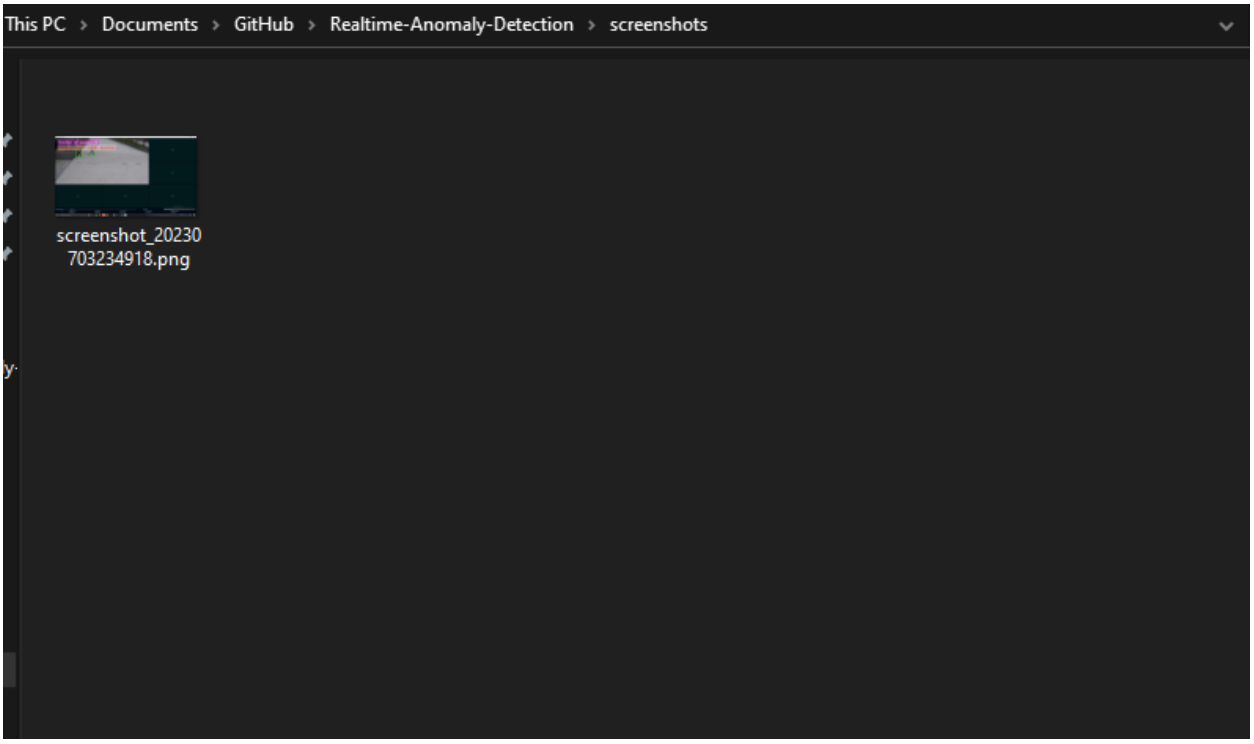
5.5 Full Program Running

When the anomaly surveillance system is running in its entirety, the user interface presents a comprehensive view of the system's components and functionalities. The screen showcases a real-time display of the camera feeds, providing visual coverage from multiple camera sources. Alongside the camera feeds, a toolbar offers a range of options for enhanced control and customization.



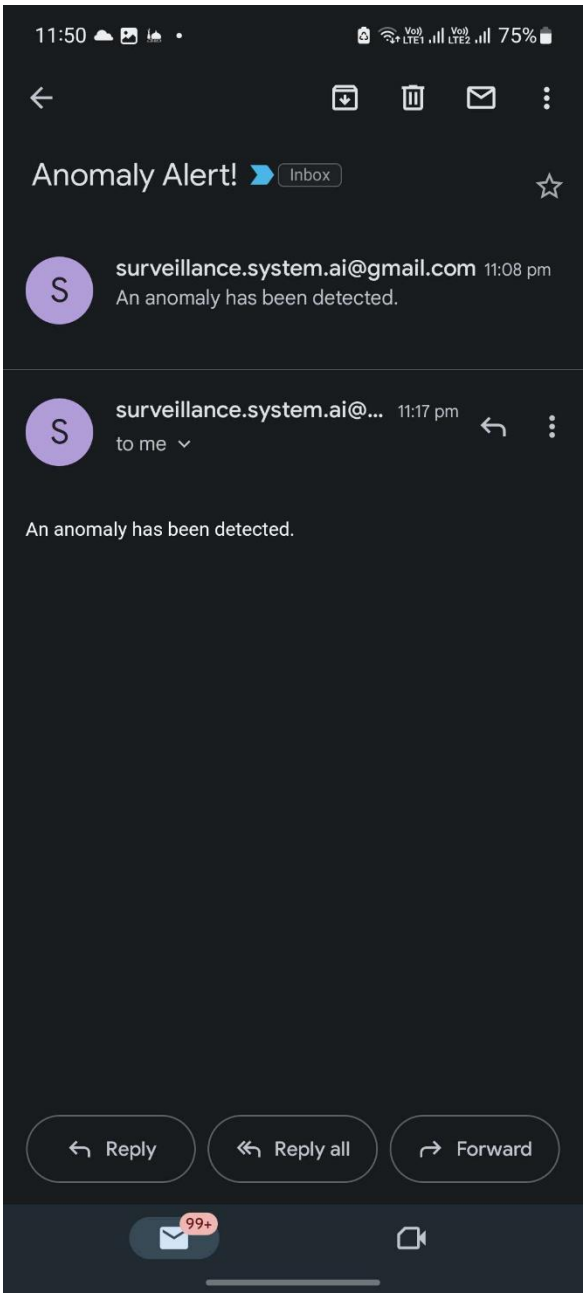
5.6 Screenshot of The Anomaly is Saved in a Folder

When an anomaly is detected within the anomaly surveillance system, the system automatically captures a screenshot of the anomaly event. This screenshot is then promptly saved in a designated folder specifically created for storing anomaly-related images. By organizing the screenshots in a dedicated folder, users can conveniently access and review the captured visual evidence, facilitating further analysis, investigation, and documentation of detected anomalies.



5.7 Notification is Sent via Email

When an anomaly is detected within the anomaly surveillance system, the system promptly generates a notification to inform relevant parties. This notification is seamlessly sent via email, ensuring immediate and widespread dissemination of information. By utilizing email as the communication medium, the system enables efficient and reliable delivery of notifications to designated recipients, enabling swift response and action in response to detected anomalies.



Chapter 6: Conclusion and Future Work

6.1 Conclusion

In conclusion, the real-time anomaly surveillance system presented in this thesis addresses the need for proactive monitoring and detection of anomalies in diverse environments. By leveraging cutting-edge technologies, such as deep learning, unsupervised learning, and multiple-instance learning, the system achieves accurate and efficient anomaly detection. The integration of user-friendly features, such as a customizable toolbar, adjustable anomaly sensitivity, and email notifications, enhances the system's usability and adaptability to different monitoring scenarios.

The system's architecture, consisting of the RTFM method and the I3D model, demonstrates its capability to handle large-scale video data and extract meaningful features for anomaly detection. Through extensive experimentation and evaluation using the ShanghaiTech dataset, the system showcases its effectiveness in detecting various anomalies, ranging from abnormal events to rare occurrences.

The visual walkthrough of the system highlights its intuitive interface and key features, including the toolbar options, the normal and anomaly states, and the saving of anomaly screenshots in a dedicated folder. The system's ability to send notifications via email further enhances its practicality for real-world surveillance applications.

Overall, the developed real-time anomaly surveillance system proves its potential to significantly enhance situational awareness, improve security measures, and provide timely responses to anomalous events. Future research can focus on expanding the system's capabilities, such as integrating additional anomaly detection algorithms, supporting more camera sources, and further optimizing performance to handle real-time processing of high-resolution video feeds.

6.2 Future work

6.2.1 Introduction

The AI Surveillance System app is designed to provide real-time video surveillance with object detection capabilities using YOLO (You Only Look Once) model. The system analyzes multiple camera feeds and identifies the number of people present in each frame. Additionally, it can determine whether an anomaly is detected based on predefined thresholds.

6.2.2 Feature Work and Potential Enhancements

- Improved Anomaly Detection:

Currently, the system uses a simple anomaly scoring method to determine if an anomaly is present. An enhancement can involve implementing more sophisticated anomaly detection algorithms, such as deep learning-based anomaly detection models, to achieve better performance and accuracy in anomaly detection.

- Multi-class Object Detection:

Expanding the object detection capabilities to detect multiple classes of objects, such as vehicles, animals, or specific items of interest, would make the system more versatile and useful in different surveillance scenarios.

- Real-time Object Tracking:

Integrating object tracking algorithms like Kalman filters or Hungarian algorithm-based methods will enable the system to track detected objects across consecutive frames, providing smoother and more consistent object trajectories.

- Camera Calibration and Perspective Correction:

Implementing camera calibration and perspective correction techniques can rectify distortions in the video feeds caused by the camera's perspective, ensuring accurate object detection and measurements in real-world coordinates.

- Cloud Integration and Remote Monitoring:

Incorporating cloud storage and remote monitoring features would allow users to access the surveillance system from anywhere, providing convenience and scalability to the application.

- Alerting Mechanism:

Adding an alerting mechanism, such as email or SMS notifications, for detecting critical events like unauthorized access, intrusion, or abnormal behavior will enhance the system's security capabilities.

References

[1]: Wikipedia (Supervised Learning)
https://en.wikipedia.org/wiki/Supervised_learning

[2]: Wikipedia (Unsupervised Learning)
https://en.wikipedia.org/wiki/Unsupervised_learning

[3] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1386–1393, 2014.

[4]: Radu Tudor Ionescu, Sorina Smeureanu, Bogdan Alexe, and Marius Popescu. Unmasking the abnormal events in video. In Proceedings of the IEEE International Conference on Computer Vision, pages 2895–2903, 2017.

[5]: Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12173–12182, 2020.

[6]: Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10076–10085, 2020.

[7]: Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. arXiv preprint arXiv:2005.02359, 2020

[8]: Yuanhong Chen, Yu Tian, Guansong Pang, and Gustavo Carneiro. Unsupervised anomaly detection with multi□scale interpolated gaussian descriptors. arXiv preprint arXiv:2101.10043, 2021.

[9]: van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1705–1714, 2019.

[10]: Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7842–7851, 2019.

[11]: Xinyang Feng, Dongjin Song, Yuncong Chen, Zhengzhang Chen, Jingchao Ni, Haifeng Chen. Convolutional Transformer based Dual Discriminator Generative Adversarial Networks for Video Anomaly Detection.

[12]: Wikipedia (Weak Supervison) https://en.wikipedia.org/wiki/Weak_supervision

[13]: Guansong Pang, Chunhua Shen, and Anton van den Hengel. Deep anomaly detection with deviation networks. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 353 – 362, 2019.

[14]: Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6479–6488, 2018.

[15]: Yu Tian, Gabriel Maicas, Leonardo Zorron Cheng Tao Pu, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Few-shot anomaly detection for polyp frames from colonoscopy. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VI 23, pages 274–284. Springer, 2020.

[16]: Peng Wu, jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In European Conference on Computer Vision (ECCV), 2020.

[17]: Muhammad Zaigham Zaheer, Jin-ha Lee, Marcella Astrid, Arif Mahmood, and Seung-Ik Lee. Cleaning label noise with clusters for minimally supervised anomaly detection. arXiv preprint arXiv:2104.14770, 2021.

[18]: Wikipedia (Multiple instance Learning) https://en.wikipedia.org/wiki/Multiple_instance_learning

[19]: Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W. Verjans, Gustavo Carneiro. Weakly-supervised Video Anomaly Detection with Robust Temporal Feature Magnitude Learning arXiv:2101.10030.

[20]: Joao Carreira, Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset arXiv:1705.07750.