8-15-2023 10:30 AM

Weakly-Supervised Anomaly Detection in Surveillance Videos Based on Two-Stream I3D Convolution Network

Sareh Soltani Nejad,

Supervisor: Haque, Anwar, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

A widespread adoption of city surveillance systems has led to an increase in the use of surveillance videos in order to maintain public safety and security. This thesis tackles the problem of detecting anomalous events in surveillance videos. The goal is to automatically identify abnormal events by learning from both normal and abnormal videos. Most previous works considered any deviation from learned normal patterns as an anomaly. However, this may not always be valid since the same activity could be normal or abnormal under different circumstances. To address this issue, this thesis utilized Two-Stream Inflated 3D (I3D) Convolutional Networks to extract spatial and temporal video features and demonstrated how it outperformed the 3D Convolutional Network (C3D) used in prior work as a feature extractor. To avoid annotating abnormal activities in training videos, a weakly supervised anomaly detection model was implemented based on the Multiple Instance Learning (MIL) framework. The model considers normal and abnormal videos as bags and video clips as instances. It learns a ranking model to predict high anomaly scores for video clips containing anomalies. The thesis further shows that the choice of features input, such as concatenating RGB and Flow features, and careful choice of optimization settings, such as optimizer, can significantly improve the performance of the anomaly detection model on some evaluation metrics.

Keywords: Anomaly Detection, weakly-supervised learning, Multiple Instance Learning (MIL), Deep Learning, Feature Extraction

Summary for Lay Audience

Anomaly detection in computer vision is the task of recognizing rare or abnormal events or behaviors in videos. This includes the presence of unexpected objects, changes in expected motion patterns, or deviations from the norm. Video anomaly detection has many applications, including medical imaging, traffic monitoring, and surveillance. Anomaly detection in surveillance videos is a vital tool for identifying potential security threats and alerting security personnel to act. However, traditional video analysis methods rely on human monitoring, which can be error-prone and time-consuming. Therefore, developing an automatic video anomaly detection system is crucial to reduce human resources and improve detection accuracy. The development of a video anomaly detection system involves feature extraction from video data. This is where appearance-based and motion-based features are identified and selected to differentiate normal behavior from abnormal behavior. Machine learning models are then trained using these features to identify anomalous behavior in video data. Recent advancements in deep learning have led to the emergence of novel methods for anomaly detection in surveillance videos, potentially achieving superior performance compared to traditional machine learning systems. In this thesis, we proposed an anomaly detection system based on both appearance-based and motion-based features to detect anomalies happened in surveillance videos. Our model demonstrated promising results in detecting abnormal events in videos.

Acknowledgements

I extend my sincere gratitude to my supervisor, Dr. Anwar Haque, whose invaluable guidance and support have been fundamental throughout my research journey. His profound knowledge, patience, and consistent encouragement have played a pivotal role in shaping my ideas and providing me with the necessary resources to successfully complete this thesis.

I am also deeply appreciative of my parents, Sakineh and Majid, along with my sister, Saeedeh, for their unwavering support not only throughout my thesis journey but also in all aspects beyond. Their belief in my potential has been a constant wellspring of motivation and resilience, even across distances. Their emotional and mental support have been invaluable, reminding me that love and care know no barriers. Thank you, Sakineh, Majid, and Saeedeh, for being my rock and my inspiration.

I want to express my gratitude to my dear friends, particularly Narges, Chandrika, Saba, Mansi, Ali Tafakor, Nima, Ali Ghavam, Mahdiyar, and Mehrdad. Their unwavering presence during the most challenging phases of my thesis journey has been immeasurable. Without their consistent mental and emotional support, the successful completion of my thesis would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Anomaly detection is one of the most complicated challenges in computer vision [10,36,41,45, 56,66,75,80]. Video anomaly detection is an area of research that concentrates on recognizing uncommon or abnormal behaviors or incidents in videos. It includes a wide range of events such as the presence of unexpected objects or incidents, a person falling, a car crash, a medical emergency, changes in the expected motion pattern, and other deviations from the standard, which are illustrated in Figure 1.1.

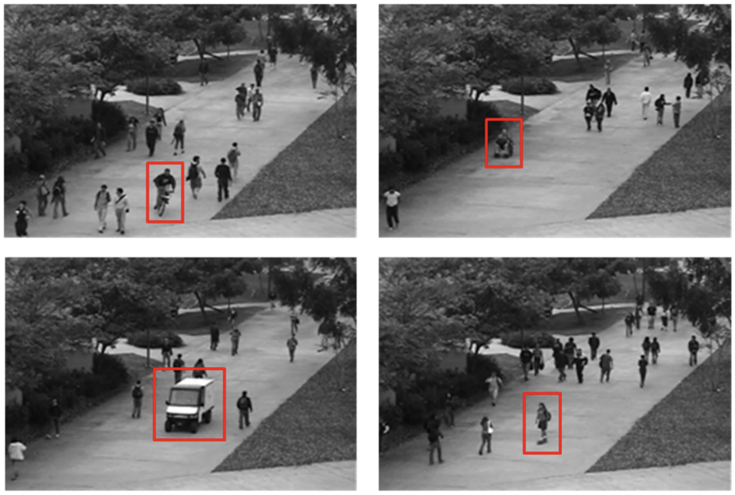


Figure 1.1: Several instances of anomalies are present in the UCSD anomaly detection dataset [41].

Anomaly detection in videos is rapidly expanding and has diverse applications such as medical imaging, traffic monitoring, and surveillance. As abnormal occurrences are rare, it can be challenging for people to recognize them while watching videos. Therefore, developing a method to detect patterns that differ from normal ones is crucial. Traditional video analysis methods depend on human monitoring, which can be error-prone and and time-consuming [12].

Anomaly detection systems are designed to detect anomalies in video footage and alert users to their presence as soon as possible. This can be beneficial in various situations, including surveillance, industrial monitoring, and other scenarios where it is crucial to detect abnormalities.

The rising demand for urban security has caused an increase in surveillance videos used in urban environments to observe human activity and prevent abnormal events. Essentially, detecting anomalies in surveillance videos is a vital tool for security and surveillance, as it helps detect potential security threats and alerts security personnel to take action. Typically, continuous monitoring by trained personnel for abnormal events in surveillance videos is labor-intensive and time-consuming. Therefore, research efforts in automatic video anomaly detection are essential to reduce the human resources required for video monitoring and improve detection accuracy.

To develop a video anomaly detection system, the initial step is to extract relevant features from the videos. Feature extraction is a critical aspect of this process, as it involves identifying and selecting significant patterns and attributes from the video data, which can be used to differentiate normal behavior from abnormal behavior. This requires analyzing the video data and identifying specific characteristics such as appearance-based and motion-based features. Appearance-based features are derived from an object's visual appearance, such as color, texture, and shape. They can be used to detect unusual behavior in video data based on changes in object appearance over time. In contrast, motion-based features are derived from the speed, direction, and acceleration of an object's motion. These features can be used to detect abnormal movements or sudden changes in motion that may indicate an anomaly.

In recent years, researchers have been utilizing various techniques, such as CNNs, VGG architectures, and C3D models, for extracting features from visual data. However, a common challenge encountered by these methods is their limited capacity to effectively capture temporal features within videos, which are crucial for tasks like anomaly detection. In our study, our primary focus was on addressing this challenge through innovative strategies. Specifically, we explored the utilization of a two-stream Inflated 3D CNN as our chosen feature extractor. This novel approach enables the extraction of both appearance-based (RGB) and motion-based (flow) features from video data. We hypothesized that this combined approach could lead to enhanced outcomes, given that the integration of both appearance-based and motion-based features offers a more comprehensive representation of video content. As a result, we expected an improvement in the accuracy and reliability of video anomaly detection systems.

Following feature extraction from video data, these features are used to train machine learning models to accurately detect any anomalous behavior in the video data. In this regard, a variety of techniques have been developed for detecting anomalies. Recent advancements in deep learning have led to the emergence of new methods for anomaly detection in surveillance videos. When more data is utilized, deep learning methods have the potential to outperform conventional machine learning systems [81].

Many existing approaches to anomaly detection assume that anomalies are deviations from a learned normal pattern. However, this assumption may not hold in the case of surveillance videos, which capture complex real-world anomalies that cannot be constructed from normal activities [66]. Furthermore, it is impossible to enumerate all possible normal activities that can be considered normal under different conditions, as some activities may be normal or abnormal depending on the context [12].

1.1 Thesis contributions

To address the limitations of previous works mentioned earlier, the principal accomplishment of this thesis is the implementation of an anomaly detection system capable of detecting anomalies with minimal supervision and less reliance on prior information. This thesis takes the following approaches:

- The proposed approach in this thesis involves a two-stream based anomaly detection system for videos. This system utilizes a two-stream Inflated 3D (I3D) Convolutional Neural Network to extract RGB and Flow features from the video. The RGB stream extracts information related to the appearance of objects and scenes, while the optical Flow stream captures the motion and dynamics of objects between frames. We then combined the information from both streams by concatenating the learned RGB and Flow features. This provides a more complete understanding of the video content, leading to improved anomaly detection accuracy.
- We extended and enhanced the anomaly detection model proposed in [66] using the PyTorch framework [57]. The detector is trained to detect anomalies using videos that are weakly labeled as normal or abnormal. We utilized the Multiple Instance Learning (MIL) framework to devise a weakly-supervised approach based on video-level annotations. We treated anomaly detection as a regression problem. The MIL framework is employed to assign a higher anomaly score to videos that are expected to contain anomalies.
- We evaluated our proposed method on the UCF-Crime [66] dataset and conducted experiments to assess its effectiveness in detecting anomalies. The results of these experiments demonstrate that our model performs well and is effective in detecting anomalies in the surveillance videos.

1.2 Thesis outline

We organize the rest of this thesis as following:

Chapter 1 introduces the research problem and proposed approach. Chapter 2 provides background information on relevant concepts. Chapter 3 reviews existing work on feature extraction techniques and video anomaly detection methods. Chapter 4 presents our proposed anomaly detection model. Chapter 5 discusses evaluation metrics and presents experimental results. Finally, Chapter 6 provides conclusions and avenues for future research that can build upon our work.

Chapter 2

Background

This chapter provides a brief summary of the relevant background topics related to the thesis. It is divided into two main sections. Section 2.1 will go through the concept of deep learning as well as the deep learning related techniques used to develop the anomaly detection system. Feature extraction techniques will be discussed in Section 2.2.

2.1 Machine Learning

Machine learning is a subset of artificial intelligence that involves training machines to learn from data and use that knowledge to make decisions or predictions.

The goal of the training process is to minimize the difference between the predicted outputs and the actual outputs of the model, which is achieved by adjusting the model parameters. This difference, also known as error, is determined using a loss or objective function, which takes in the parameter values as input and attempts to find the optimal parameter values that minimize the error [65].

Machine learning can broadly be categorized into three main subcategories: Supervised Learning, Unsupervised Learning, and Weakly-Supervised Learning. Supervised learning involves training the model with labeled data, whereas unsupervised learning involves using unlabeled data to identify patterns and relationships. In weakly supervised learning, some part of the training is labeled and the remainder of the data is either unlabeled or weakly labeled. In this research, our model is based on weakly supervised learning. As a result, a brief explanation of this approach is provided in the following section.

2.1.1 Weakly Supervised Learning

Weakly Supervised Learning (WSL) [87] is a form of machine learning in which training data is not completely labeled. WSL is a type of supervision in which only a subset of the training data is labeled and the remainder of the data is either unlabeled or weakly labeled. This can happen when obtaining completely labeled data is costly or time-consuming, or when the data is intrinsically ambiguous or difficult to label. Unlike fully supervised learning, where a model is trained with a comprehensive set of labeled data, WSL utilizes a set of weak or incomplete labels. These labels are image tags, text documents, or a set of data points that belong to a certain class, but not all of them. WSL aims to improve the performance of models trained on a smaller quantity of labeled data by leveraging the large amount of available unlabeled data.

Weakly supervised learning has gained popularity in recent years due to the availability of large-scale datasets and the cost and time associated with manual annotation. Unlike fully supervised learning, where the model is trained on fully labeled data, in weakly supervised learning, the model is trained on data that is partially annotated or has weak annotations such as image tags, image-level labels, or bounding boxes. This enables the learning process to be performed on a much larger scale, as annotating large-scale datasets can be time-consuming and costly [66].

WSL has a variety of practical applications, including computer vision, natural language processing, and bioinformatics, where acquiring a huge amount of labeled data can be difficult and resource-intensive. In computer vision, WSL can be applied to training object detection models using image tags or bounding boxes rather than full object annotations. In natural language processing, using a small set of labeled data and a large set of unlabeled data, WSL can be applied to train models for text classification or sentiment analysis. In bioinformatics, WSL can be utilized to train models with limited labeled data to find new drugs, predict how proteins work, and study how genes are expressed.

There are several techniques used in WSL, such as self-training and multiple instance learning. These techniques attempt to leverage weak labels to learn a model that can generalize to new data, even in the face of incomplete or noisy labels.

Self-training [84] is a common method for training models without enough annotated data in weakly supervised learning. It is a type of semi-supervised learning in which a model is trained on a small amount of labeled data and then used to predict the labels of the rest of the unlabeled data. The predicted labels are then added to the labeled data to form a new, more extensive labeled dataset. This procedure is repeated multiple times until a suitable result is obtained.

According to [85], self-training is used to improve the model's performance by leveraging a large amount of unlabeled data. It has been demonstrated that self-training is effective for different computer vision and natural language processing tasks. As highlighted in [85], self-training has been employed in object detection, text classification, and semantic segmentation to improve the performance of models trained on limited annotated data.

Multiple Instance Learning (MIL) is a weakly supervised learning methodology [87]. Based on [50], MIL is a sort of learning paradigm in which the training data comprises sets of instances known as ”bags,” where each bag is associated with a single class label but the instances within the bag may or may not be labeled. In fact, the purpose of MIL is to learn a topic using both positive and negative bags of instances. Each bag may contain multiple instances, but a bag is considered positive when only a single instance meets the idea. On the other hand, a bag is only designated negative if every instance contained within it is negative [50]. MIL is beneficial when obtaining instance-level labels is difficult or costly, while bag-level labels are readily available.

MIL aims to train a model that can predict the class label of a new set of instances, given only the bag-level labels for the training data [8]. Figure 2.1 presents a comparison between supervised learning and multi-instance learning, where the classifier is trained using bags of instances instead of individual instances [38]. According to [8], the main idea behind MIL is to use bag-level labels to deduce the class labels of instances within the bag and then build a model that can generalize to new instances. MIL is used for a variety of computer vision tasks where annotations are limited or difficult to acquire. For example, the goal of object detection is to detect objects in an image, and the annotations are often image-level labels indicating the presence or absence of objects in the image, rather than instance-level annotations indicating the location and shape of objects in the image. In addition, MIL has been utilised in bio-informatics to estimate the function of a protein based solely on gene expression data.

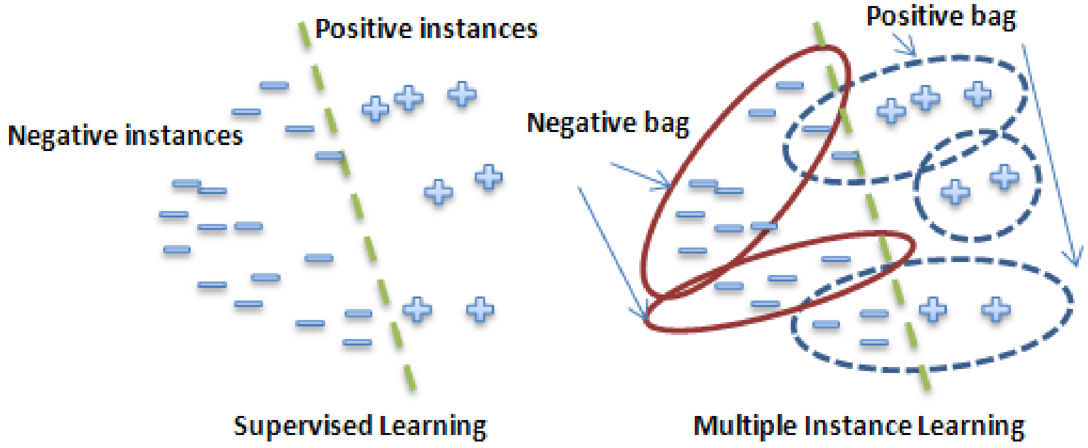


Figure 2.1: Comparison of supervised learning and multi-instance learning [38]

There are two main approaches to solving the MIL problem: instance-level approaches and bag-level approaches. Instance-level approaches model the relationship between the instance features and the instance labels, while bag-level approaches model the relationship between the bag features and the bag labels. Instance-level approaches require more annotations than bag-level approaches, but they provide more precise predictions.

2.2 Deep Learning

Deep learning is a type of machine learning inspired by the human brain's adaptability and can perform human-like tasks with greater accuracy. It is more powerful than other machine learning approaches because it can automatically and optimally extract features as part of the learning process [65]. Deep learning methods are based on neural networks, which process input data through computational layers to produce classification outputs. Each layer consists of interconnected neurons, which combine data from the previous layer using weighted connections. The activation function of each neuron evaluates the weighted sum of inputs to determine its effect on later layers. During training, the network performs a forward pass for each data sample, and the weights are adjusted to optimize a loss function after each pass.

Deep Neural Networks (DNNs) are neural networks that have many computational layers between an input layer and an output layer. These computational layers, which are also known as hidden layers, are where learning takes place. By having multiple hidden layers, DNNs can learn from multiple levels of abstraction [24]. Figure 2.3 illustrates the structure of a deep neural network.

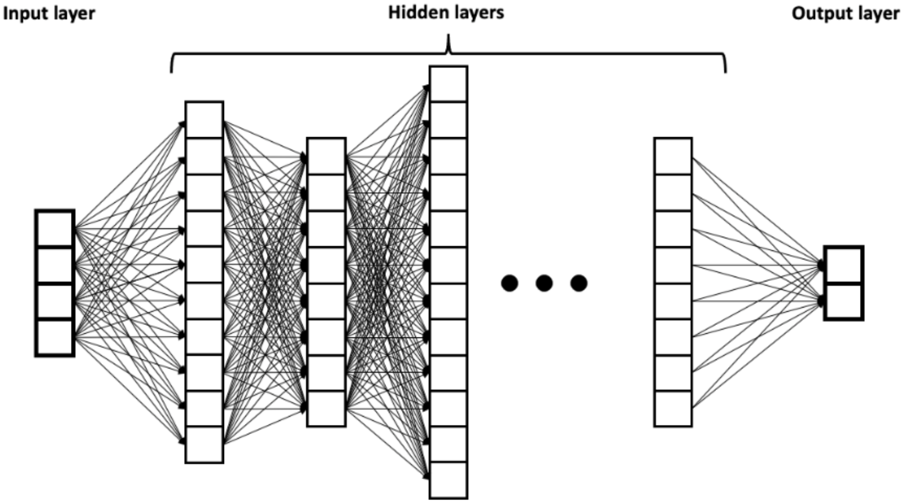


Figure 2.2: Deep Neural Network Architecture [2]

2.2.1 Convolutional Neural Networks

Convolutional neural networks (CNN) are a type of deep neural network that processes visual data such as images and videos. CNNs can learn various levels of abstraction from an input image at different layers. The first layers usually learn basic features like edges and colors, whereas deeper layers extract more complex features such as shapes and objects [24]. A CNN is composed of several layers, including an input layer, convolutional layers, pooling layers, fully connected layers, and an output layer. During training, the input layer receives a batch of images, where each image has specific dimensions for width, height, and channel size. **Convolutional Layers:** A convolutional layer in a CNN uses a kernel to perform element-wise multiplication on local regions of the input image and produce a feature map that extracts specific features from the image [24].

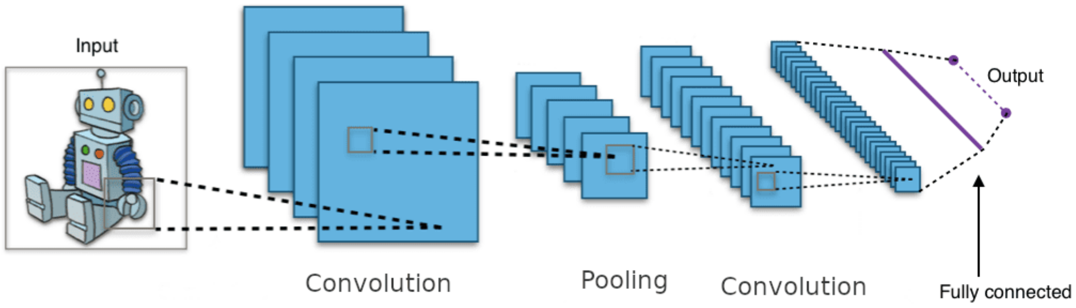


Figure 2.3: Convolutional Neural Network Architecture [1]

Activation Function: An activation function is commonly applied after each convolutional layer. This function takes the sum of the values of each element in the filter and applies a non-linear transformation to produce an output. Each element in the filter is a weight that is multiplied by a corresponding element in the input image, within the receptive field of the filter [79]. Activation functions used in CNNs add non-linearity, which is essential for updating the weights after each forward pass through the network. This process is called backpropagation, which uses the chain rule to find the partial derivatives or gradients of the loss or objective function with respect to each weight. When calculating the gradients, the derivatives of the activation functions also must be considered. Linear activation functions do not provide non-linearity and result in a constant factor during weight updates, preventing any significant improvement in the network output. The rectified linear unit (ReLU) is a widely used non-linear activation function in CNNs because it is simple, fast, and produces a more predictable gradient during back-propagation compared to other non-linear activation functions.

Pooling Layer:

CNNs use pooling layers to reduce data dimension by down-sampling it. This is achieved by either averaging or finding the maximum value in each region of the feature map from previous layers and passing the resulting value to the next layer. The process of taking the average of a region is called average pooling, while taking the maximum value of a region is called max pooling.

Fully Connected Layers: CNNs' fully connected layers make the final classification decision. Flattening the final convolutional layer's output into a vector feeds one or more fully-connected layers. The output of each neuron in a fully connected layer is the weighted sum of the inputs followed by an activation function. The fully connected layers link the convolutional layers' high-level features to class scores, which indicate the probability of each input belonging to a particular class.

Output Layer: In the last layer of a CNN, the model produces a final classification. The raw output values are usually processed through a SoftMax function, which normalizes them to real values between 0 and 1. These normalized values are considered probabilities for each class, and the class with the highest probability is chosen as the final output. SoftMax is typically used for multi-class classification, and it is similar to the sigmoid function used in binary classification.

The formula for SoftMax is as follows:

$$\sigma(x_i) = \left(\frac{e^{x_i}}{\sum\limits_{j} e^{x_j}} \right) j = 1, ..., n \tag{2.1}$$

where x_i is the i_th element of the input vector, N is the size of the input vector, e is the base of the natural logarithm, and sum(e(̂x_j)) is the sum of the exponential values of all elements in the input vector.

2.3 Feature Extraction from videos

Feature extraction is a process used in computer vision and machine learning to extract important information from video streams, which can be used for various purposes such as classification, tracking, and action detection. The choice of features for video anomaly detection depends on the type of analysis required. In this process, spatial and temporal features are usually used. These features help detect abnormal behavior in a video by capturing color information and motion patterns, which can be indicative of suspicious activities or events.

2.3.1 Representing videos as RGB and Flow features

This section discusses the two main types of features used in computer vision: spatial and temporal. Spatial features, such as RGB, provide information about the visual characteristics of an object or region of interest, including its location, shape, color, and texture. These features are commonly used in tasks such as recognizing objects and separating them from the background. On the other hand, temporal features, such as flow, capture object movement in video. They are computed based on the apparent motion of pixels between consecutive frames. These features are useful for tasks such as recognizing actions, detecting events, and summarizing videos. Combining both spatial and temporal features can provide a more complete representation of the video content, which is useful in various tasks such as action recognition, object tracking, and anomaly detection. By combining both types of features, we can gain a better understanding of the content and movement in the video, leading to more accurate analysis and detection of significant events.

2.3.2 Features Extraction Techniques

Techniques for Extracting Spatial Features

The widespread popularity of Convolutional Neural Networks (CNNs) has been propelled by their remarkable ability to extract spatial features from images and videos. Originally introduced in 1988 for the analysis of medical images [70], CNNs have evolved significantly since their inception, overcoming early limitations rooted in processing power and dataset availability. The surge of deep learning and computational advancements has elevated CNNs to a pivotal role in an array of image and video tasks, encompassing areas like face detection, speech recognition, image classification, and recommendation systems [81]. Diverging from traditional neural networks, CNNs employ convolutions instead of matrix multiplications, thus diminishing the weight count and overall network complexity. An intriguing facet is that raw images can be directly input into the network [81], bypassing the need for feature extraction inherent in conventional learning methods. The architectural design of CNNs adeptly capitalizes on spatial relationships, leading to a reduction in network parameters [81], while optimization through back-propagation algorithms further augments their performance. As we delve into the subsequent sections, we explored several advancements beyond CNNs that have emerged in the quest for improved feature extraction methodologies.

VGG which stands for Visual Geometry Group, represents a widely recognized deep Convolutional Neural Network (CNN) architecture distinguished by its multiple layers. The term ”deep” indicates the considerable number of layers, with VGG-16 and VGG-19 featuring 16 and 19 convolutional layers, respectively. K. Simonyan and A. Zisserman [63] created the VGG16 Convolutional Neural Network model. This model achieved a test accuracy of 92.7% in ImageNet, which is a dataset containing over 14 million images categorized into 1000 classes [16]. The structure of VGG16 is depicted in Figure 2.4 [19]. It consists of thirteen convolutional layers and three fully connected layers. All of the hidden layers have a nonlinearity function called rectification (ReLU). Each of the thirteen convolutional layers can be divided into five different blocks [81]. The first block has two distinct layers with 64 channels. Block 2 has two convolutional layers with 128 channels. The number of channels in Block 3 is increased from 256 to 512 [81]. The final two blocks have three convolutional layers each with 512 channels. After each block, there is a max-pooling layer with a size of 2x2. After the sixth block, three fully connected layers are added, the first two of which have a total of 4096 channels, and the third layer has 1000 channels. The number of channels in the fully connected layers can be adjusted according to the specific requirements of different datasets. The last layer is called Soft-Max. The VGG19 model adheres to the same foundational structure as VGG16, albeit with an extension to accommodate 19 layers. The numerical labels ”16” and ”19” correspond to the count of weight layers within the model, specifically the convolutional layers. As a result, VGG19 incorporates an additional set of three convolutional layers in comparison to its predecessor, VGG16. The VGG model improves on Convolutional Neural Networks by introducing two key changes. The first change is to reduce the convolutional kernel size to 3x3.

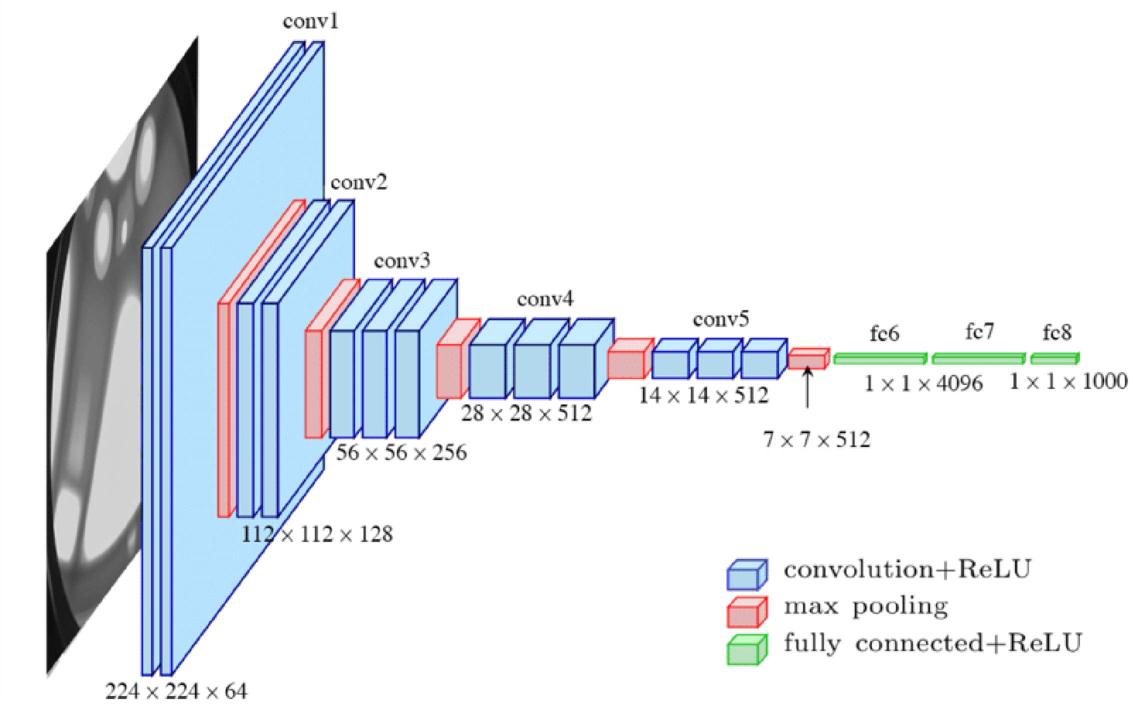


Figure 2.4: The structure of VGG16 network [19]

This reduces the amount of computation and the number of parameters required compared to larger kernels. The second change is to increase the depth of the CNN, showing that deeper networks can achieve better performance. These improvements make VGG a highly effective CNN model for image classification tasks.

Dense Convolutional Neural Network (DenseNet): A DenseNet is a type of Convolutional Neural Network that uses interconnected layers to enhance its functionality [31]. While traditional CNNs aimed to increase network depth and layer size, DenseNet prioritizes feature development. This approach allows for a more efficient and effective network.
DenseNet architecture is composed of several dense blocks, each of which contains multiple convolutional layers, as depicted in Figure 2.5. Unlike VGG16, where layers within a block are not interconnected, layers in the same block in DenseNet are connected, with each layer receiving the output features of the layers above it. DenseNet typically utilizes small convolutional kernels and employs an 11 Convolutional layer after each average pooling layer to act as a transition between two dense blocks. Compared to other models, DenseNet requires fewer feature images, as each layer in the network receives feature maps from all previous layers, allowing for a more compact network. Furthermore, the dense blocks collect more data, reducing the number of parameters and feature images required to maintain the stability of the entire training phase.

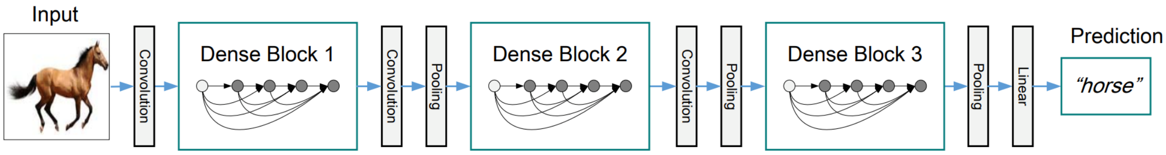


Figure 2.5: A DenseNet model with three dense blocks [31].

Techniques for Extracting Temporal Features

Recurrent Neural Network (RNN): Recurrent Neural Networks (RNNs) are specifically designed for sequential or time-series data processing tasks, such as Natural Language Processing (NLP) and video processing. They are characterized by their memory feature, which allows past inputs to affect the current output [32]. Figure 2.6 demonstrates the difference between RNNs and other types of networks. The input layer's value is represented by x, and o represents the output layer's value. The hidden layer consists of three value-handling matrices, namely U, V, and W. In traditional Neural Networks, U and V directly process each input x, and the resulting structure produces the output. Therefore, x1 contributes to output O1. However, RNNs utilize an additional matrix W to manage the values of previous hidden layers. As a result, the subsequent output O2 is still influenced by the preceding input x1, allowing RNNs to maintain a memory of previous inputs [81].

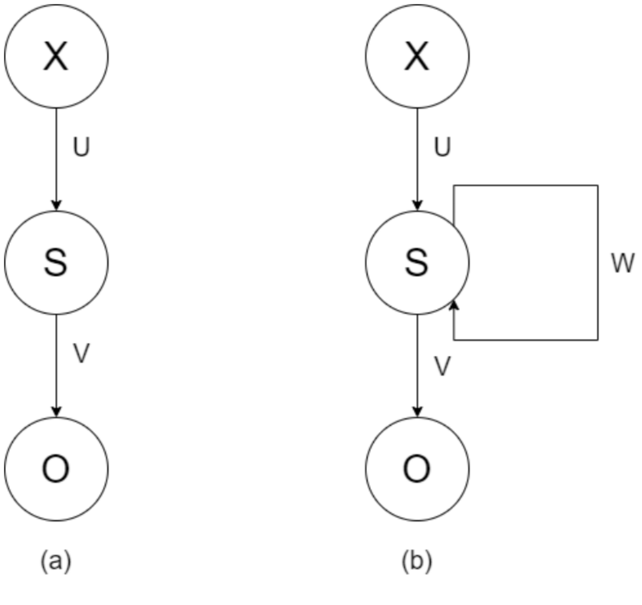


Figure 2.6: (a) Traditional Neural Network Architecture (b) The Structure of RNN [81]

Long Short-term Memory (LSTM): Recurrent Neural Networks (RNNs) are well-suited for processing sequential data, but their basic structure is prone to issues like gradient disappearance or explosion when working with excessively large sequential data. To address this, Long Short-term Memory (LSTM) has emerged as a replacement solution to enhance the fundamental structure of RNNs [29]. In an RNN, only the hidden state S_t is passed to the following moment. However, LSTM introduces a new state, called the cell state (C_t), which carries information across time steps. The LSTM structure is illustrated in Figure 2.5 [29]. Each LSTM unit contains gates that determine whether the input should be retained, updated, or passed through. As a result, LSTM can handle inputs and hidden states, effectively addressing the limitations of conventional RNNs.

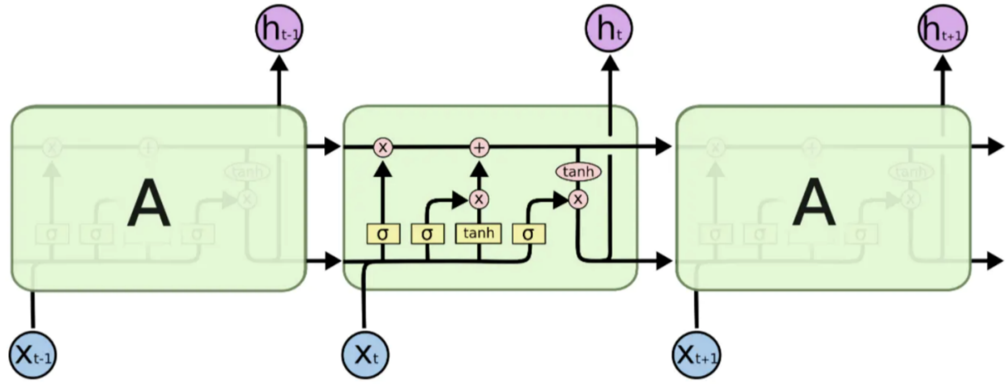


Figure 2.7: LSTM Structure [29]

Two-stream Based Model: The two-stream model is an alternative approach to RNNs and LSTM for temporal feature extraction, with a focus on capturing motion information and temporal dynamics. The structure of the two-stream model for temporal feature extraction is shown in Figure 2.8 [62]. This model consists of two streams, the spatial and the temporal stream. The spatial stream takes a randomly sampled video frame as input and uses a simple CNN model with four convolutional layers and two fully connected layers to extract spatial features that are crucial for image classification tasks. On the other hand, the temporal stream uses a Convolutional Neural Network structure and takes as input the optical flow displacement fields between frames. Optical flow captures horizontal and vertical motion information in the video, which enables the extraction of temporal features from this stream. Compared to RNNs and LSTMs, which attempt to extract information from sequential data, the two-stream model separates the spatial and temporal aspects of the data, allowing it to capture motion information and temporal dynamics effectively.

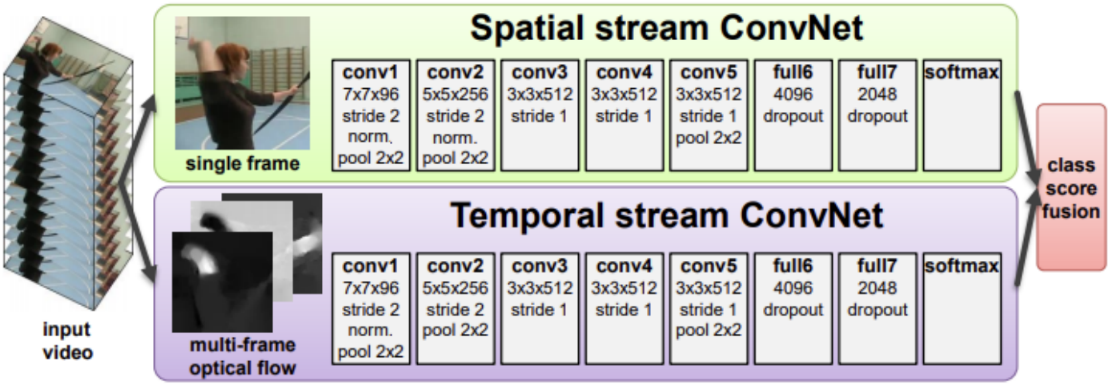


Figure 2.8: Two-stream structure for video classification [62]

Convolutional 3D (C3D): Traditional convolutional neural networks (CNNs) are limited in their ability to capture temporal information in videos as they are designed to focus on static images or short sequences of images. To address this limitation, 3D convolutional networks (C3D) have been developed as an extension of CNNs. Unlike 2D convolutional kernels which produce 2D feature maps irrespective of whether they are applied to static or video images, 3D kernels can extract temporal features due to the additional temporal dimension [69]. The initial structure of the C3D model is shown in Figure 2.9. The architecture is similar to a 2D CNN, but it is designed to extract temporal features from video data. The model consists of eight convolutional layers and two fully connected layers. The 3D convolution kernels used in the model are set to 3 x 3 x 3, corresponding to the length (number of input frames), height, and width of the input data, respectively. The pooling kernels have a size of 2 x 2 x 2. The first convolutional layer of the model has 64 filters, and the number of filters in each subsequent convolutional layer is doubled after each pooling layer [69].

C3D inherits some advantages from traditional 2D CNNs. For example, all model kernels are identical, which reduces computational load. The model extracts local features, and as the number of convolutional layers increases, global features can be extracted. However, because of the additional temporal dimension, the C3D structure is more complex, and training is more challenging. Describing the features of each 3D convolutional layer can be difficult, which makes C3D a black box model.

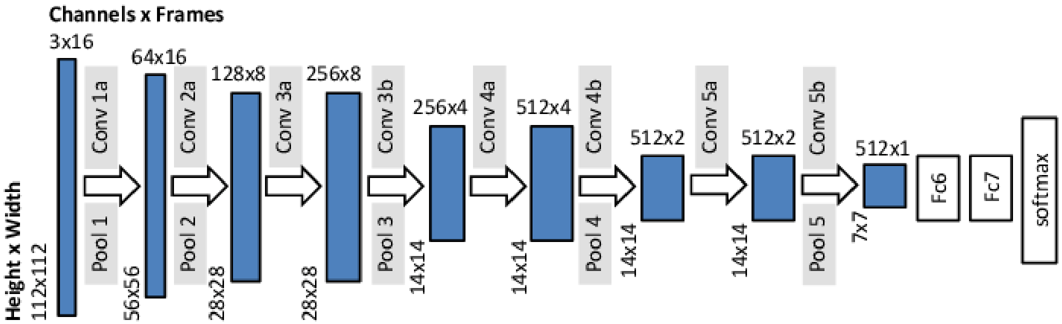


Figure 2.9: C3D architecture with eight convolution layers [18]

Inflated 3D (I3D): Inflated 3D (I3D) Convolutional Neural Network is a type of deep learning architecture used in computer vision tasks, particularly for action recognition in videos. It was proposed by Joao Carreira and Andrew Zisserman in 2016 [11], and it has since become a popular method for video analysis tasks. It is trained on large video datasets such as Kinetics 400 [33], which is an action recognition dataset of realistic action videos obtained from YouTube, containing 306,245 videos from 400 action categories. It has demonstrated strong performance in action recognition benchmark datasets and has been widely used in various applications such as video classification, human pose estimation, and activity recognition.

It is based on the concept of 2D Convolutional Neural Network inflation, where 2D Convolutional Neural Network filters and pooling kernels are extended to 3D. This extension enables the network to learn spatio-temporal feature extractors from videos. Figure 4.3 shows the architecture of I3D. The main difference between I3D and C3D is their architecture and the way they

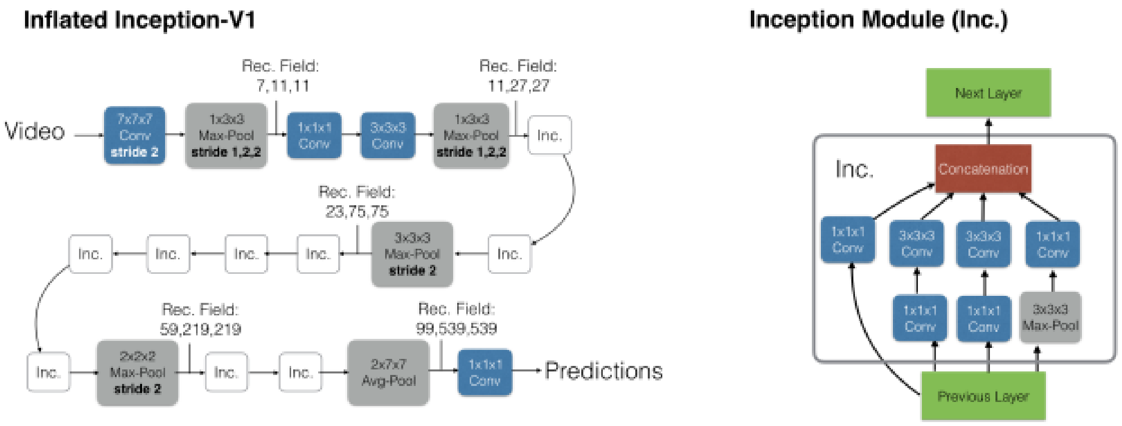


Figure 2.10: Structure of Inflated 3D Convolutional Neural Network [11]

process video data. I3D is based on the 2D Convolutional Neural Network inflation approach, where the filters and pooling kernels of 2D Convolutional Neural Networks are expanded to 3D, enabling the learning of spatio-temporal feature extractors from video. I3D networks are pre-trained on large-scale datasets such as ImageNet and Kinetics, and then fine-tuned for specific video classification tasks. The architecture of I3D is deeper than that of C3D, with more convolutional layers, and it has larger filter sizes, which allows it to learn more complex spatio-temporal features. I3D also uses a two-stream architecture, where one stream processes RGB inputs and the other stream processes optical flow inputs. This approach allows I3D to capture both appearance and motion information and has shown to provide state-of-the-art performance on many video recognition tasks.

C3D, on the other hand, is purely 3D Convolutional Neural Network, which means it processes video data directly in 3D space. It has eight convolutional layers and two fully connected layers, with all of the 3D convolutional kernels having dimensions of 3x3x3, and all pooling kernels having a configuration of 2x2x2. C3D is trained end-to-end on specific video classification tasks, without pre-training on other datasets. Unlike I3D, C3D only processes RGB inputs.

In terms of performance, I3D has shown better accuracy in several video recognition tasks compared to C3D, due to its ability to leverage pre-training on large-scale image classification datasets. Additionally, I3D is faster to train and requires less memory than C3D, as it uses pre-trained 2D models as a starting point for feature extraction.

Chapter 3

Related work

The detection of anomalies in videos is known as ”video anomaly detection.” Due to the rarity of abnormal occurrences, it might be challenging for a person to watch videos and detect the probable abnormal activities that are taking place in the movie [81]. Therefore, it is essential to develop a method for detecting video patterns that deviate from an established concept of normal patterns. Anomaly detection is the term that we use for these kinds of activities [12].

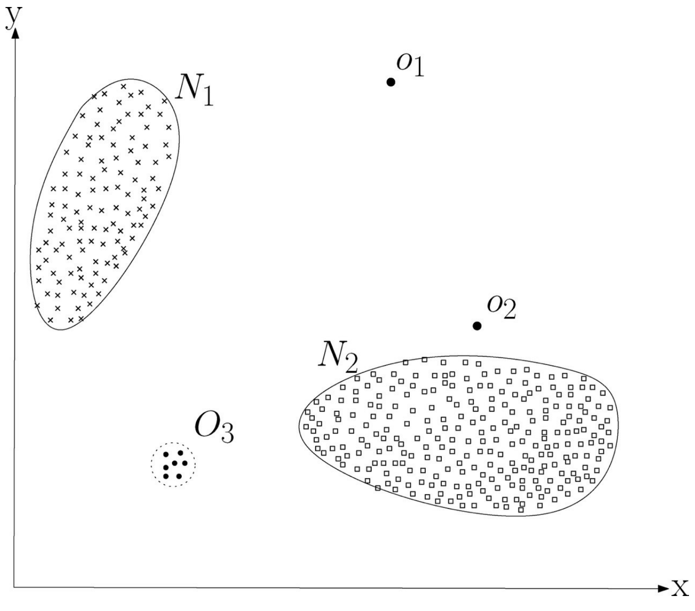


Figure 3.1: An example of anomaly [12]

A simple example of an anomaly is depicted clearly in Figure 3.1. The data exhibits a pair of normal distributions, denoted N1 and N2. Outliers, such as points o1 and o2 and those in region O3, are considered anomalies [12].

Video anomaly detection is one of the most complicated challenges in computer vision [9, 10, 36, 41, 45, 56, 66, 75, 80, 83]. In this chapter, we review how prior work addressed the problem of anomaly detection in videos.

3.1 Traditional Anomaly Detection Methods

The preliminary work in detecting anomalies in videos is founded on the assumption that anomalies arise abruptly and are rare, and any deviation from the standard pattern is considered as an abnormal event [25]. To encode normal patterns, various statistical models are used, including Gaussian process based models [13, 40], social force models [53], Hidden Markov-based models [25, 30, 37], histogram-based methods [15], motion patterns [61], mixtures of dynamic textures model [42], the spatial-temporal Markov random field based models [25, 54, 72], and context-driven method [88]. The mentioned methods consider anomalies to be outliers [25]. However, these traditional methods have limitations when it comes to analyzing large volumes of video data. They may not perform well under such circumstances and may not be scalable enough to identify outliers [81].

3.2 Deep Learning-based Anomaly Detection Methods

Recent studies have employed deep learning algorithms to address the limitations of using traditional methods in anomaly detection. Deep learning algorithms are more adept at handling such data, and have been shown to improve the performance of anomaly detection. The emergence of deep learning techniques has transformed computer vision [82, 89], and in the last few years, these methods have become the primary means of detecting anomalies in videos. These methods aim to train models to detect abnormalities based on learned features. There are three categories of deep learning-based methods: reconstruction-based, prediction-based and hybrid methods [39].

Reconstruction-based methods are the most commonly used deep learning methods for video anomaly detection, according to recent studies [4, 14, 21, 86]. These methods involve training a deep learning model to reconstruct the frames in a video, and the difference between the original and reconstructed frames is used to differentiate between normal and abnormal events. Abnormal events have a higher reconstruction error than normal events because they deviate more from the training data. The architecture of models used for video anomaly detection is similar to image-based models like CNNs, but additional methods like LSTM and 3D Convolutional Neural Networks are included to process temporal features and extend the image-based structure to videos [62]. Reconstruction-based methods, including convolutional auto-encoders, are used in studies such as [22,26,59], while [60] incorporates deep adversarial training into the reconstruction models. In the realm of video surveillance applications, various efforts have been made to detect instances of violence or aggression within videos. Kooij et al. [35] utilized both video and audio data to identify aggressive actions in surveillance videos. Mohammadi et al. [55] introduced a novel behavior heuristic-based approach for categorizing videos as either violent or non-violent. Incorporating tracking methodologies, authors in [73] introduced an alternative perspective beyond distinguishing between violent and non-violent patterns. They proposed modeling the typical motion of individuals and subsequently detecting anomalies by identifying deviations from this established normal motion. Because acquiring dependable tracks can be challenging, some methodologies circumvent tracking and instead learn overarching motion patterns using techniques like topic modeling [30], histogram-based methods [15], motion patterns [61], and Hidden Markov Model (HMM) applied to local spatio-temporal volumes [36]. Indeed, by utilizing training videos that depict typical behaviors, these approaches gain an understanding of the statistical distributions associated with normal motion patterns. Consequently, they are able to detect patterns that exhibit lower probabilities as potential anomalies. However, reconstruction-based methods focus solely on reconstructing individual frames or patches, without considering the temporal relationships between frames in a video. This approach can limit their effectiveness in detecting anomalies that occur over multiple frames or have unique temporal characteristics that cannot be captured through individual frame reconstructions.

Prediction-based methods for video anomaly detection involve using predictions of future frames based on historical data to detect anomalies in video streams. These methods generate anomalous frames with the same features as the training video using auto-encoders [44, 51]. Anomalies are anything that deviates from the forecast of a deep learning model. In studies such as [48, 49, 52], sequence models such as Convolutional LSTM (ConvLSTM) are used for future frame prediction and anomaly detection. Additionally, generative adversarial networks (GANs) are utilized for anomaly detection based on prediction in [23, 78]. Prediction-based methods take into account the temporal nature of video data, which can make them more accurate than reconstruction-based methods for detecting anomalies that occur over time. By predicting future frames based on historical data, prediction-based methods can detect subtle changes in the video stream that may not be apparent in a single frame or even a short sequence of frames.

The previously discussed methods rely heavily on prior knowledge to detect anomalies, assuming that any deviation from normal patterns is an abnormality. Furthermore, obtaining normal training data requires annotations, which is challenging and time-consuming, especially for videos. The assumption that an anomaly is a deviation from a normal pattern, as made by previous methods, may not always be valid. This is because it is difficult to define all normal patterns that can be considered normal in all situations [66]. For example, the same activity can be considered normal or abnormal depending on the context. Additionally, some anomalies can be complex and cannot be reconstructed from normal patterns [66].

To address the limitations of anomaly detection methods that rely on prior knowledge of events, it is necessary to develop methods that require minimal supervision and do not heavily rely on prior knowledge. One such approach is to treat anomaly detection as a classification or regression problem using weakly labeled training videos, which requires less annotation effort. As a binary classification task, a classifier can generate precise features for normal and abnormal videos, whereas treating it as a regression problem allows the use of an anomaly score to measure the likelihood of a video being anomalous.

Hybrid methods for detecting anomalies in videos involve training the anomaly detection model using both normal and abnormal videos. These methods use a technique called Multiple Instance Learning (MIL) to model motion patterns in a weakly supervised setting. In this setting, the model distinguishes between videos containing normal or abnormal events using only video-level labels [6, 28, 66, 83]. Sultani et al. [66] created a binary classifier based on MIL to detect anomalies. In [83], a graph convolutional neural network is utilized to clean up label noise. They approached anomaly detection as a regression problem and used a deep ranking model to predict anomaly scores in a weakly supervised setting.

In this study, we developed a weakly supervised anomaly detection system by extending the baseline approach proposed by Sultani et al. [66]. The baseline approach utilizes C3D (Convolutional 3D) [69] to extract features from videos; however, C3D only captures spatial information and does not incorporate temporal information that can be crucial for video anomaly detection. To address the limitations of the C3D feature extractor, we proposed an approach that utilizes Two-Stream Inflated 3D Convolutional Neural Network (I3D) as a feature extractor, which integrates both spatial and temporal information from the video frames. This integration of spatial and temporal features provided by I3D Convolutional Neural Networks provides a more comprehensive understanding of the video content, resulting in enhanced performance in anomaly detection tasks compared to using only temporal features extracted by C3D.

Chapter 4

Proposed Framework and Methodologies

In the previous chapter, we reviewed key prior work that addressed the problem of video anomaly detection. In this chapter, we present a two-stream based system for detecting anomalies and elaborated on how spatial and temporal features are extracted from videos. We discuss a basic approach to weakly supervised anomaly detection introduced in [66]. While we implement a similar pipeline, we propose a different approach that employs a two-stream Inflated 3D (I3D) Convolutional Neural Network to extract both RGB and Flow features from the video, instead of C3D as used in [66]. Finally, we explain how we could analyze this pipeline from various aspects such as visual inputs.

4.1 High-Level Architecture

The process of detecting anomalies in surveillance videos can be broken down into two stages. The first stage involves extracting RGB and Flow features from the videos using a feature extractor module, which creates I3D features. In the second stage, these features are used to detect anomalies in the videos. Our approach to this task involves using a two-stream I3D network to extract features and a weakly-supervised anomaly detector based on multiple instance learning to detect abnormal behaviors. The overall structure of our framework is shown in Figure 4.1.

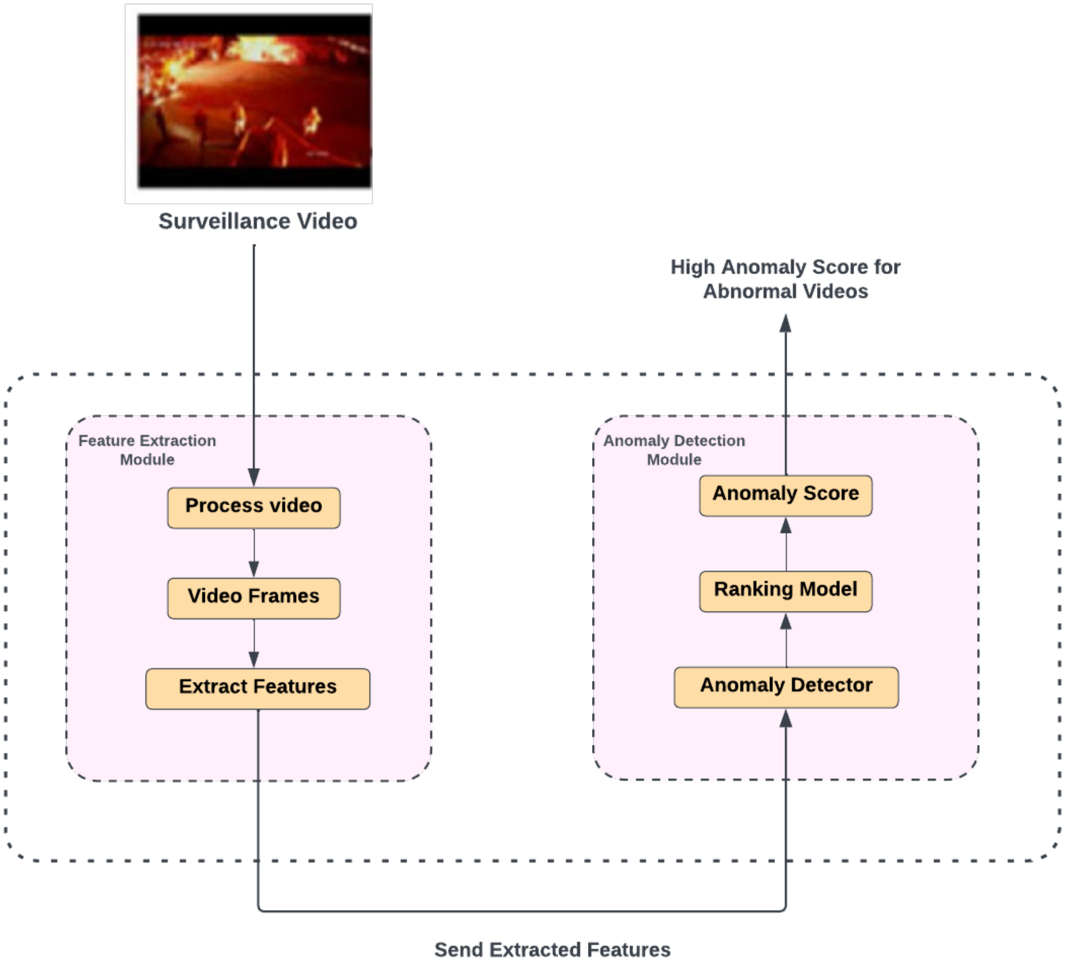


Figure 4.1: High-Level Proposed System Architecture

To begin, we extract visual features from the input video using an Inflated 3D Convolution Neural Network. These features are then passed on to an anomaly detector. We use the UCF-Crime dataset [66], which includes 1900 surveillance videos with 13 different types of realistic anomalies, as the basis for our research. In the following sections, we provide more details about our framework, which consists of two main components: a two-stream I3D network as a feature extractor and a weakly-supervised anomaly detector. We further explain how visual features are extracted from the videos and how anomalies are detected based on the extracted features.

4.2 Feature Extractor

Here we describe our input and output representation for feature extraction from videos.

4.2.1 Input/output representation

To extract features from a video, the first step is to divide it into N segments, denoted by v_i (where i = 1,2,...,N). After dividing the video into segments, the video segments are inputted into the two-stream I3D Convolutional Neural Networks. In the next section, we explained the structure of the feature extractor, illustrated in Figure 4.2.

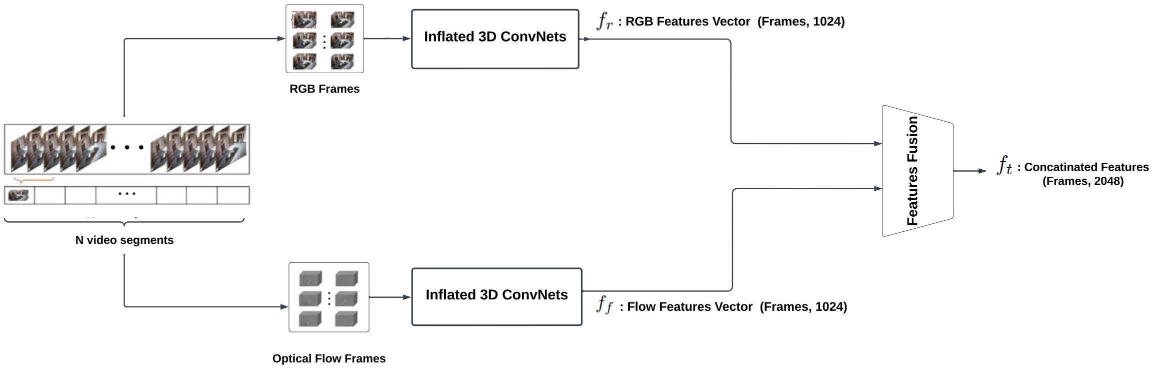
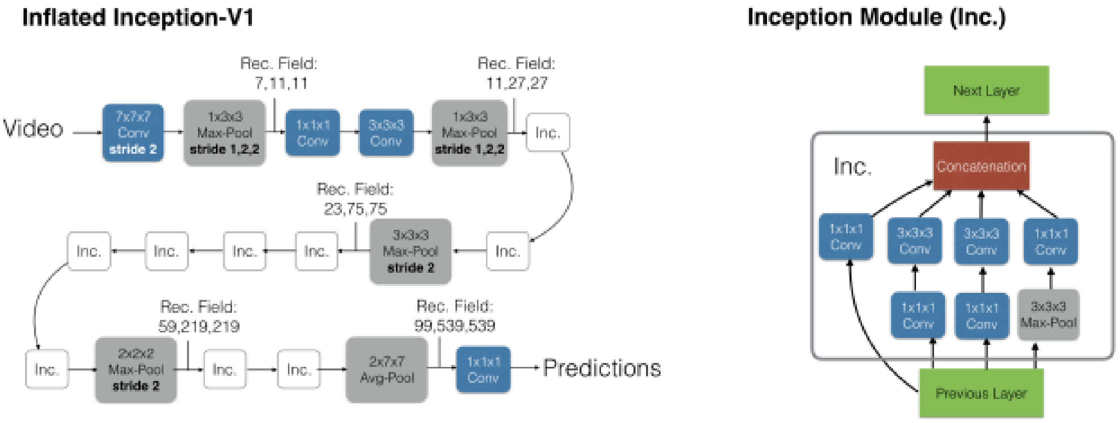

RGB Frames
Inflated 3D ConvNets
f_r : RGB Features Vector (Frames, 1024)
N video segments
Features Fusion
f_t : Concatinated Features (Frames, 2048)
Inflated 3D ConvNets
f_f : Flow Features Vector (Frames, 1024)
Optical Flow Frames

Figure 4.2: The architecture of the two-stream Inflated 3D Convolutional Neural Networks

4.2.2 Method

We utilized a two-stream Inflated 3D (I3D) Convolutional Neural Network to extract features from videos. It was trained on a large dataset called Kinetics 400 [33], which contains over 300,000 videos depicting various action categories. I3D [11] is an extension of 2D Convolutional Neural Networks commonly used for image classification. However, unlike 2D Convolutional Neural Networks that only consider spatial information, I3D considers both spatial and temporal information in videos. This is achieved by inflating 2D filters to 3D filters, allowing the network to learn spatiotemporal features from video data [11].

To elaborate, the I3D model is created by expanding a 2D convolutional neural network (CNN) called InceptionV1 [67]. This expansion includes inflating all 2D filters and pooling kernels to 3D equivalents by adding a time dimension. The model is initialized using pre-trained parameters from the Kinetics 400 dataset, which involves repeating the weights of the 2D convolution kernels N times along the time dimension and re-scaling them by dividing them by N. In order to accurately represent the spatiotemporal information in videos, it's important to balance the size of the time dimension of the 3D filters. If the time dimension is too large, it can lead to distorted edges, and if it's too small, key dynamic information may be lost. Therefore, when converting 2D filters to 3D filters, factors such as frame rate and image size must also be considered. Additionally, to preserve features extracted by the shallow network, the initial two max-pooling layers have kernels of size 1 × 3 × 3 with time dimension strides of 1, while the kernel of the last average-pooling layer has dimensions 2 × 7 × 7 with a time dimension stride of 2 [77]. Figure 4.3 illustrates the overall structure of I3D and the Inception module.


Inflated Inception-V1

Inception Module (Inc.)

Figure 4.3: The structure of I3D Convolutional Neural Network [11]

The two-stream I3D Convolutional Neural Networks is a version of the I3D architecture designed for video action detection that combines RGB and optical flow data. The structure of the proposed two-stream Inflated 3D Convolutional Neural Networks is shown in Figure 4.2.

The model consists of two parallel I3D networks, where one network is trained on RGB frames and the other on optical flow data. By using both streams, the model can extract spatial and temporal information from video data. The RGB stream captures information on the appearance of objects and scenes in a video. In contrast, the optical flow stream includes information about the motion and dynamics of objects between consecutive frames [77]. The combination of these streams allows the model to capture a more comprehensive understanding of the actions in the video.

Extracting Features from RGB Frames

The RGB stream of an I3D network consists of several blocks of convolutional and pooling layers. The convolutional layers use 3D filters that are inflated from 2D filters to capture both spatial and temporal information in the video by sliding the kernels across sequential video frames. The pooling layers reduce the spatial dimensions of feature maps while retaining temporal information. The final layer of the RGB stream is a fully-connected layer that makes predictions based on the learned features.

The extracted RGB features provide visual information about the video's content. This includes the position and shape of objects and people, as well as information about lighting and color. In summary, after processing a video segment through the RGB stream, the output is a 1024-dimensional tensor which represents the RGB features of the segment, denoted as f_r.

Extracting Features from Optical Flow Data

In order to compute Flow features in the I3D network, optical flow between consecutive video frames must be calculated first [11]. We used the TVL1 algorithm to calculate this optical flow. TV-L1 is an optical flow method that minimizes the Total Variation (TV) and the L1 norm of the flow between two frames. It balances the smoothness of the flow with the accuracy of the flow vector at each pixel, producing results that are both smooth and accurate [74]. The optical flows are then passed through the inflated 3D convolutional kernels of the I3D network to extract Flow features in the same way as RGB features, undergoing batch normalization, activation functions (such as ReLU), and pooling layers to obtain high-level, abstract features used for video classification. The extracted Flow features encompass a wide range of information, including the motion and activity of objects and individuals within the scene. Therefore, the flow stream generates a tensor with 1024 dimensions that represents Flow features for a specific video segment. This tensor is referred to as f_f.

Fusion Layer

RGB and Flow features complement each other and can be merged to produce a more comprehensive representation of video content. To achieve this, the outputs of both streams are combined in a late fusion step. This step involves concatenating the features learned from both streams to create a final prediction. Essentially, two-stream I3D produces two 1024-dimensional tensors - one for RGB features and the other for Flow features. These tensors are then combined by concatenation to create a 2048-dimensional tensor that contains both RGB and Flow features. This concatenated tensor, denoted by f_t is used as input for the anomaly detection model, as described in the next section.

4.3 Weakly-Supervised Anomaly Detection Model

4.3.1 Input/output representation

As described in the previous section, the concatenated feature vector, denoted by f_t, which includes both RGB and Flow features, for every video segment was inputted into the anomaly model. In the following section, we described an anomaly detector that detects anomalies given vector features from the first stage, as summarized in Figure 4.4.

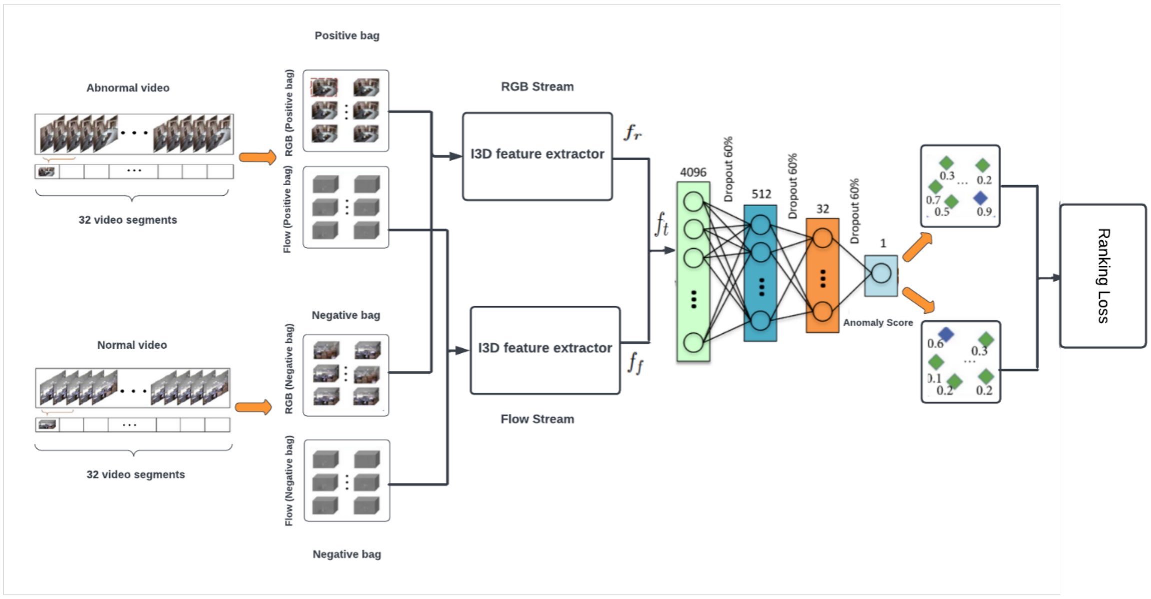


Figure 4.4: The architecture of the proposed anomaly detection model

Method

This thesis aims to develop a model for detecting abnormal events or incidents, which only requires weakly labeled data. We extended the anomaly detection model proposed in [66] as our baseline anomaly detector based on PyTorch [57]. The detector was trained to detect anomalies using surveillance videos with weakly labeled data. This implies that the videos are classified as normal or abnormal, without specifying the exact locations of the anomalies. Therefore, the model relies on video-level labels instead of instance-level labels to detect anomalies. To leverage video-level annotations in weakly-supervised methods, we used the Multiple Instance Learning (MIL) framework [50].

Multiple Instance Learning is a type of learning approach where the training data consists of groups of instances called ”bags,” where each bag is assigned to a single class label but the instances within the bag may or may not be labeled. MIL is used to learn a concept using both positive and negative bags of instances. A bag is classified as positive if at least one instance is positive. Conversely, a bag is classified as negative if all instances within it are negative [50]. To apply MIL approach to our problem, we divided each video into N segments, each of the segments was defined by v_i (where i = 1,2,...,N), and each video was represented by $V = \{v\}_i^N$. Therefore, we considered each video as a bag and each video segment as an instance within the bag. If a video has at least one segment that contains anomalies, it is labeled a positive bag and denoted by B_a. Conversely, if all segments in the video are normal, it is considered a negative bag and denoted by B_n. We then extracted I3D features from each video segment of both positive and negative bags by using the two-stream I3D network, described in Fiqure 4.3.

Once we extracted I3D features for the video segments, we trained a three-layer fully connected neural network to assign an anomaly score to each segment.

Ranking Model: In our approach, we addressed the challenge of limited abnormal videos for training, as well as the time-consuming task of annotating segment-level labels, by treating anomaly detection as a regression problem instead of a classification problem.

We employed a deep ranking loss, as proposed in [66], to generate high anomaly scores for abnormal segments compared to normal segments, such as:

$$f(v_a) > f(v_n) \tag{4.1}$$

In Equation 4.1, $f(v_a)$ refers to the predicted anomaly score for an abnormal video segment, and $f(v_n)$ refers to the predicted score for a normal video segment. Due to the lack of explicit annotations for video segments, we computed the ranking loss between the highest scored instances in the positive bag, denoted by B_a, (segments containing anomalies) and the negative bag, denoted by B_n, (segments without anomalies). In this regard, the above ranking function 4.1 is modified as follows:

$$\max_{i \epsilon B_a} f(v_a^i) > \max_{i \epsilon B_n} f(v_n^i) \tag{4.2}$$

where $f(v)$ denotes the predicted score for given video segment. The purpose of this approach is to train the anomaly detector to distinguish between abnormal and normal segments even in the absence of explicit annotations.

The ranking loss between the top-scoring instances in the positive bag and the negative bag is computed using the hinge-loss function, as stated in [66]. The formula for the hinge-loss function is derived from equation 4.2 and is as follows:

$$l(B_a, B_n) = \max(0, 1 - \max_{i \epsilon B_a} f(v_a^i) + \max_{i \epsilon B_n} f(v_n^i) \tag{4.3}$$

Due to the short duration of anomalies, their corresponding anomaly scores in video segments should be sparse. To address this issue, the sparsity constraint is incorporated into the loss function for this task. The smoothness constraint is also introduced to make sure that the change in anomaly score is gradual across adjacent segments, which is important because video is a sequence of segments [66]. Together, these constraints improve the algorithm's ability to detect anomalies in short video segments. Therefore, the resulting loss function is given by:

$$l(B_a, B_n) = \max(0, 1 - \max_{i \epsilon B_a} f(v_a^i) + \max_{i \epsilon B_n} f(v_n^i) \tag{4.4}$$

$$+\lambda_1 \sum_{i}^{n} f(v_a^i) + \lambda_2 \sum_{i}^{n-1} (f(v_a^i) - f(v_a^(i + 1)))^2 \tag{4.5}$$

Where λ_1 represents the sparsity parameters, and λ_2 denotes the smoothness parameter. For better performance, we set them to 0.00008 as mentioned in [66].

Chapter 5

Experimental Setup and Results

In this chapter, we explain our experimental setup including the dataset, implementation details, evaluation metrics and quantitative experiment results, respectively.

5.1 UCF-Crime dataset

We chose to conduct our experiments using the UCF-Crime dataset [66] for specific reasons, given its distinct advantages over other available datasets. This dataset encompasses extended surveillance videos that portray a wide array of complex real-world anomalies. Unlike many alternative anomaly detection datasets, the UCF-Crime dataset offers a substantial collection of videos with longer durations. Moreover, its anomalies are intricately designed to closely resemble real-world scenarios, making it highly suitable for detecting sophisticated anomalies in surveillance videos. The dataset contains a significant amount of video content, totaling 128 hours, thereby providing a comprehensive representation of diverse and intricate anomalies that occur in complex environments. Table 5.1 presents a comprehensive comparison of existing anomaly detection datasets [66].

Dataset	Dataset length	Number of videos	Average number of frames per video
UMN [64]	5 min	5	1290
Avenue [46]	30 min	37	839
UCSD Ped1 [41]	5 min	70	201
UCSD Ped2 [41]	5 min	28	163
Subway Entrance [5]	1.5 hours	1	121,749
Subway Exit [5]	1.5 hours	1	64,901
UCF-Crime [66]	128 hours	1900	7247

Table 5.1: A comparison between anomaly datasets [66].

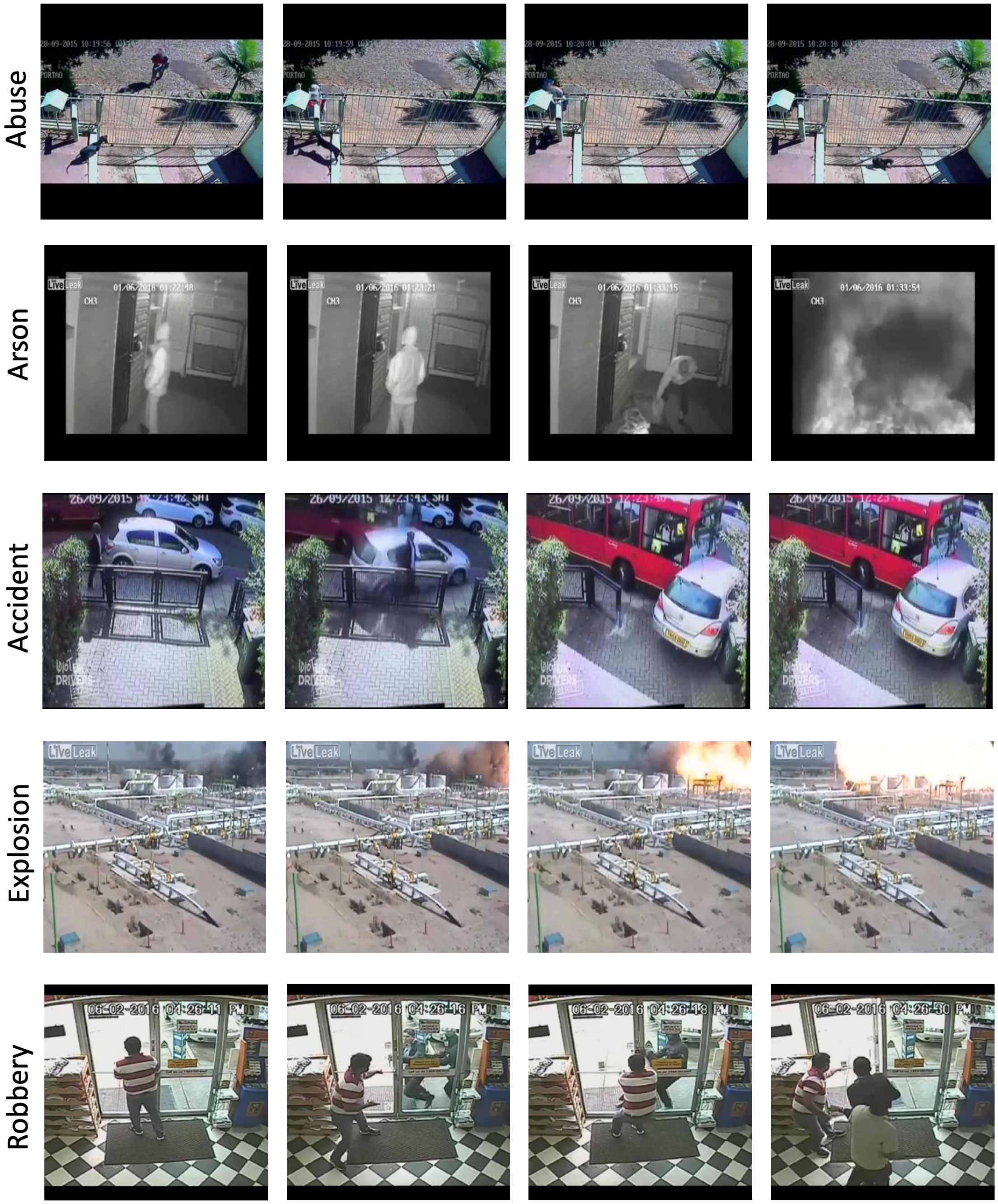
5.1.1 Dataset statistics

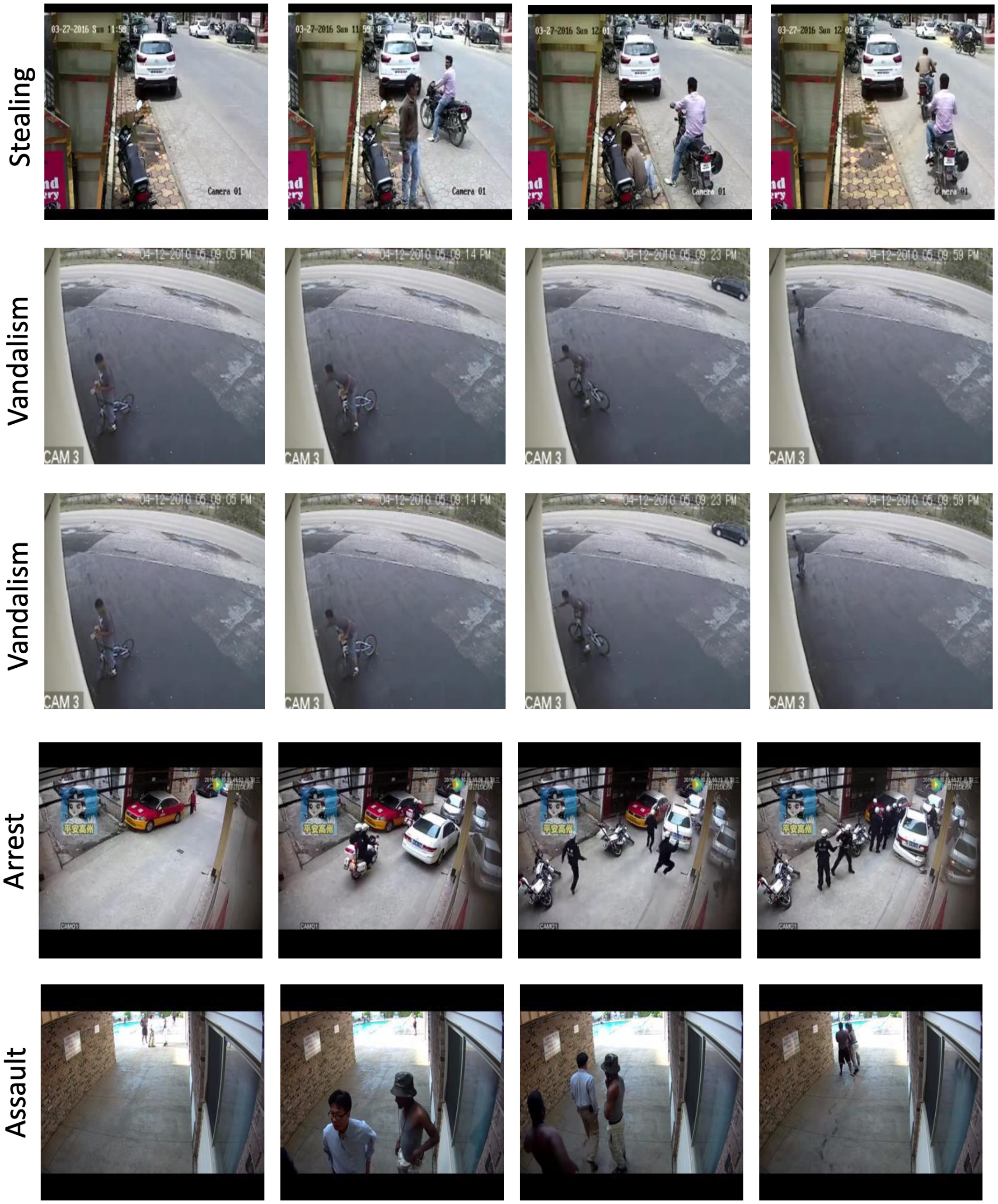
The UCF-Crime dataset [66] contains 128 hours of video. It comprises 1900 lengthy, unedited, real-world surveillance videos with 13 realistic anomalies [66]. On average, there are 950 videos containing real-world anomalies and 950 normal videos. The number of instances of anomalies in each class is presented in Table 5.2.

Anomaly	Numbers of videos	Anomaly	Numbers of videos
Abuse	50	Robbery	150
Arrest	50	Shooting	50
Arson	50	Shoplifting	50
Assault	50	Stealing	100
Burglary	100	Vandalism	50
Road Accidents	150	Explosion	50
Fighting	50	Normal events	950

Table 5.2: The total number of videos associated with each anomaly in the UCF-Crime dataset [66].

The UCF-Crime dataset includes multiple illustrations of abnormalities extracted from the training and testing videos. Some of these examples are exhibited in the figures below.


Abuse
Arson
Accident
Explosion
Robbery


Stealing
Vandalism
Vandalism
Arrest
Assault

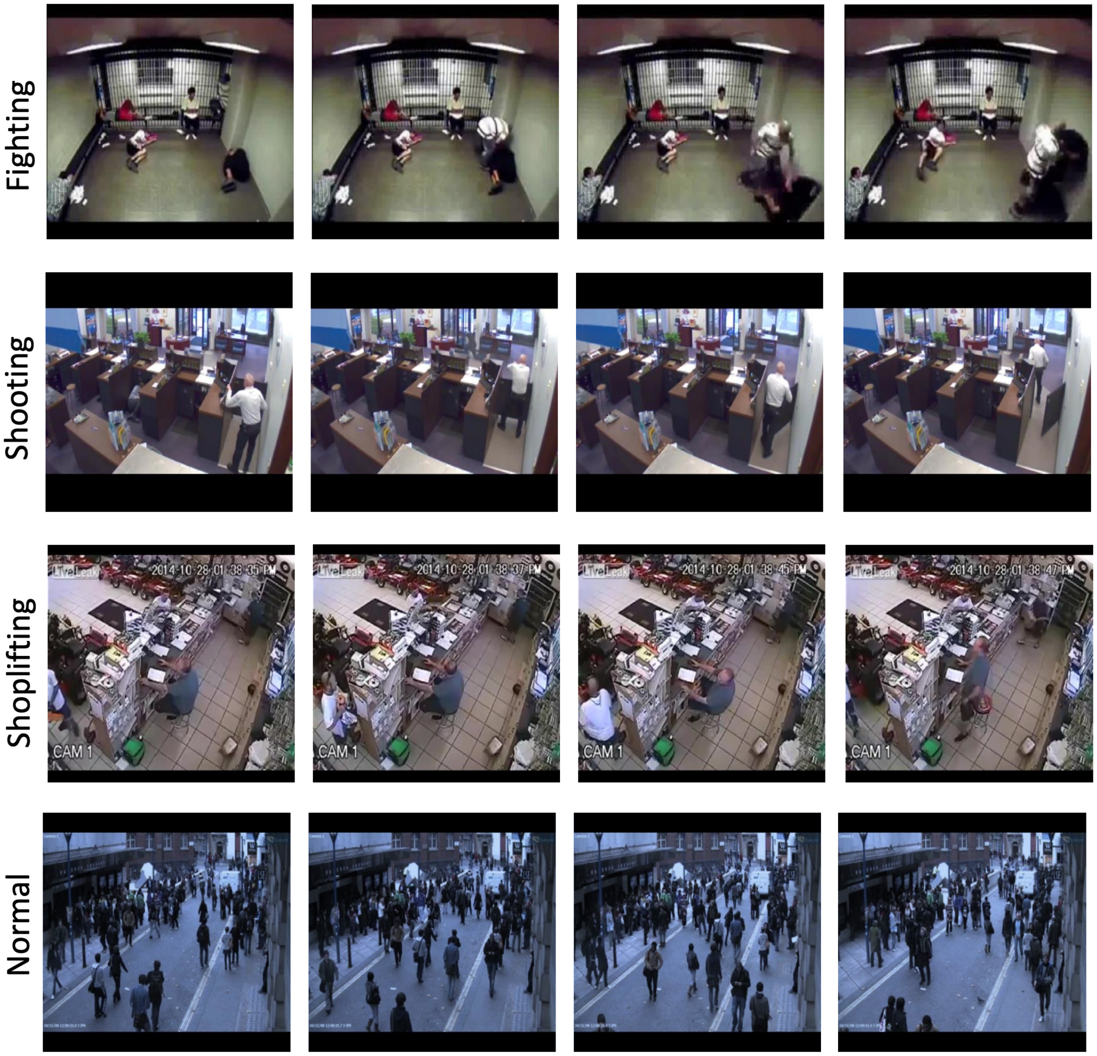


Figure 5.1: Illustrations of anomalies extracted from the UCF-Crime dataset [66].

5.1.2 Class mappings

We adopted the UCF-Crime dataset mapping. There are 13 classes for anomalies and one class for normal videos, which are used to train the anomaly detection model. These 13 categories for anomalies are Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Table 5.3 displays a comparison of anomaly classes present in various datasets.

Dataset	Anomaly classes
UMN [64]	Run
Avenue [46]	Run, throw, new object
UCSD Ped1 [41]	Bikers, small carts, walking across walkways
UCSD Ped2 [41]	Bikers, small carts, walking across walkways
Subway Entrance [5]	Wrong direction, No payment
Subway Exit [5]	Wrong direction, No payment
UCF-Crime [66]	Abuse, arrest, arson, assault, accident, burglary, fighting

Table 5.3: Comparison of anomaly classes. Numbers from the [66] paper.

5.1.3 Train and test splits

The UCF-Crime dataset consists of 1900 surveillance videos, with the training set consisting of 800 normal videos and 810 abnormal videos. The testing set includes 150 normal videos and 140 anomaly videos, covering all 13 anomaly categories in both sets. In our experiment, we utilized the default class mapping of the UCF-Crime dataset to facilitate comparison with the baseline results.

5.2 Implementation Details

We implemented our model with PyTorch [57]. To create the proposed model, we first extracted visual features, including RGB and Flow features, from the last layer of the I3D network [11]. To accomplish this, we divided each video into 32 segments, each containing 16 consecutive frames. We then fed each video segment into the proposed two-stream Inflated 3D Convolutional Neural Network (I3D) network, trained on the Kinetics dataset. From the RGB and Flow streams, we generated two 1024-dimensional tensors representing the RGB and Flow features for each video frame. We then calculated the average of the extracted features for all 16 frames within each video segment to compute the features for that segment. The RGB stream provides information about the visual appearance of objects and scenes in a video, such as color, texture, and shape. On the other hand, the optical flow stream provides information about the motion and dynamics of objects, such as the direction, speed, and acceleration of movement within the video frames. By combining these two streams of information, a more comprehensive understanding of video can be achieved. We concatenated RGB and Flow features as input to our model.

After concatenating the RGB and Flow features, resulting in a 2048-dimensional tensor, this tensor is fed into a three-layer fully connected (FC) neural network as described in [66]. The first FC layer of this network has 512 units, followed by a second FC layer with 32 units, and a final FC layer with only 1 unit. We recreated the anomaly detector proposed in [66] using PyTorch and considered it as our baseline. In our experiments, we set the batch size to 30, and randomly selected 30 videos from both the abnormal and normal video datasets to train the model. We trained the model with the Adagrad [17] optimizer using the 0.001 learning rate.

5.3 Hyper-parameter Tuning

As part of our model optimization process, we conducted a hyper-parameter search for optimizers and learning rates. Specifically, we considered two of the most popular optimization algorithms in deep learning, namely Adam [34] and Adagrad [17]. We tested these optimizers with learning rates selected from a grid of [0.01, 0.001, 0.0001].

Upon analyzing the results, illustrated in Figure 5.2 and Figure 5.3, we observed that Adagrad outperformed Adam in detecting anomalies in our model. We attribute this superior performance to the fact that Adagrad has lower complexity than Adam. Adagrad uses a per-parameter adaptive learning rate, which allows it to automatically adjust the learning rate of each parameter based on its previous gradients. This results in a more efficient and accurate optimization process, particularly when dealing with sparse data or high-dimensional feature spaces.

Overall, our hyperparameter search suggests that Adagrad may be a more effective optimizer than Adam for anomaly detection tasks, particularly in complex deep learning models.

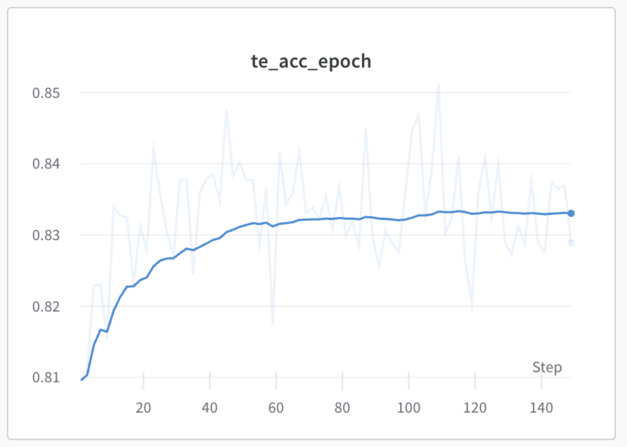


Figure 5.2: The AUC results based on Adagrad Optimizer

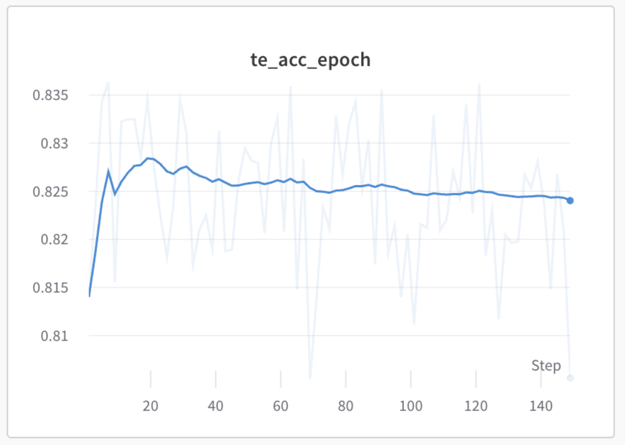


Figure 5.3: The AUC results based on Adam Optimizer

5.4 Evaluation metrics

In this section, we delve into the assessment of the proposed anomaly detection model's performance using the UCF-Crime dataset as the foundation. The selected evaluation metric for this study is the Area Under the Curve (AUC) derived from the Receiver Operating Characteristic (ROC) curve, the most common metric used in previous research on video anomaly detection [7, 27, 43, 44, 47, 58, 66, 68, 71, 76]. In our proposed approach, we treated anomaly detection as a regression task, focusing on ranking instances based on their likelihood of being anomalies. Traditional classification metrics such as F1 score, Precision, and Recall may not align with our proposed ranking-based anomaly detection system [20]. To elaborate, the proposed anomaly detection system assigns a ranking or score to each instance to indicate its anomaly likelihood. The final decision on classifying an instance as an anomaly depends on applying a threshold to these scores, leading to different levels of Precision and Recall. The F1 score overlooks the significance of the threshold at which anomalies are identified, making it less relevant for ranking-based systems. Moreover, obtaining accurate anomaly labels can be challenging in real-world situations, making precise calculation of metrics like Precision, Recall, or F1 score difficult.

Given these challenges, it is more appropriate to use evaluation metrics that are specifically designed for ranking anomaly detection tasks. The Area Under the ROC Curve (AUC-ROC) is considered a good evaluation metric for ranking-based anomaly detection tasks for the following reasons:

- **Threshold Independence:** The AUC-ROC metric is not influenced by a particular threshold value. It assesses how well a model can differentiate between different classes without needing a specific threshold to be defined. This characteristic is well-suited for our proposed anomaly detection approach, where we treated anomaly detection as a regression task and assigned high anomaly scores to videos that contain anomalies.
- **Comprehensive Assessment:** The AUC provides an aggregate measure of performance across all possible thresholds. Since the AUC considers performance over the entire

range of thresholds, it provides a comprehensive view of the model's ability to rank instances correctly, regardless of where the threshold is set. This property is especially valuable in situations where the optimal threshold might vary based on the specific needs of the application or the trade-offs between false positives and false negatives. The ROC curve, which plots the true positive rate against the false positive rate at different thresholds, provides a graphical representation of the trade-off between true positive rate and false positive rate as the threshold changes.

In conclusion, while F1 score, Precision and Recall is useful metrics for traditional classification tasks, it might not capture the complexities and priorities of our proposed anomaly detection system that relys on ranking scores. Using metrics tailored to ranking-based tasks like AUC can provide a more accurate and informative evaluation of the system's performance. The AUC-ROC metric serves as the benchmark to gauge the efficacy of our proposed approach, allowing us to both evaluate its effectiveness and make comparisons with other state-of-the-art methods in the anomaly detection domain.

5.4.1 ROC-AUC

The Area Under the Curve (AUC) of the Receiver Operating Character- istic (ROC) curve is a performance evaluation metric commonly used in binary classification problems. It plots the True Positive Rate (TPR) versus the False Positive Rate (FPR) at different classification thresholds to assess a classification model's ability to discriminate between positive and negative classes, as demonstrated in Figure 5.4. The true positive rate (TPR) is the ratio of correctly predicted positive instances (true positives) to the total actual positive instances. On the other hand, the false positive rate (FPR) is the ratio of incorrectly predicted positive instances (false positives) to the total actual negative instances. TPR and FPR are calculated as follows:

$$TruePositiveRate = \frac{TP}{TP + FN} \tag{5.1}$$

$$FalsePositiveRate = \frac{TN}{FP + TN} \tag{5.2}$$

The AUC provides an aggregate measure of performance across all possible thresholds. A perfect classifier would have an AUC of 1, indicating a TPR of 1 and an FPR of 0 across all possible threshold values. A random classifier would have an AUC of 0.5, indicating no ability to distinguish between the two classes. Therefore, the higher the AUC, the better the performance of the classifier.

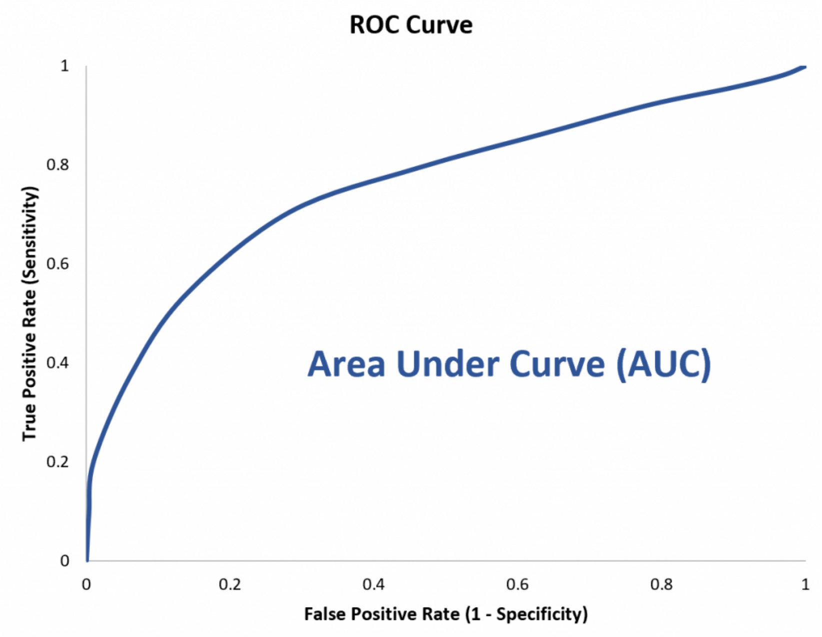


Figure 5.4: The Receiver Operating Characteristic (ROC) Curve [3]

5.5 Experimental Results

In this section, we assess how well the model performs depending on the type of video feature extraction network employed. We evaluated the model's performance by calculating the AUC for three different scenarios. Firstly, we presented the results obtained using the I3D RGB stream network, as depicted in Figure 5.5. Secondly, we evaluated anomaly detection performance using the I3D flow stream network, as illustrated in Figure 5.6. Finally, we calculated the AUC for the two-stream network, which fuses both RGB and flow stream networks, illustrated in Figure 5.7. Moreover, the model loss during the training phase for the two-stream network is also calculated and presented in Figure 5.8.

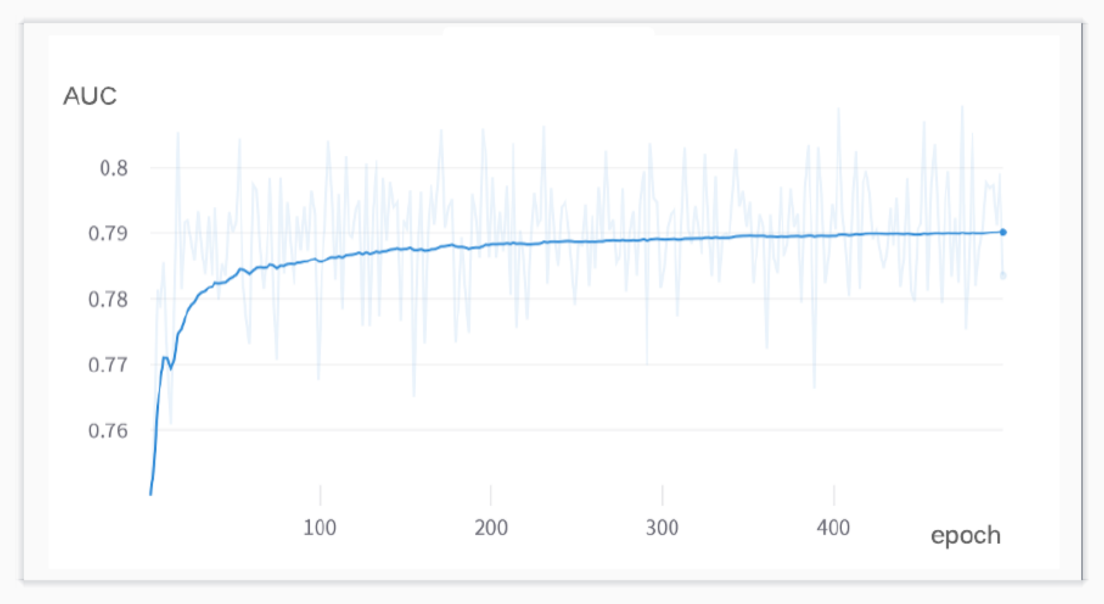


Figure 5.5: The AUC results based on I3D RGB stream network

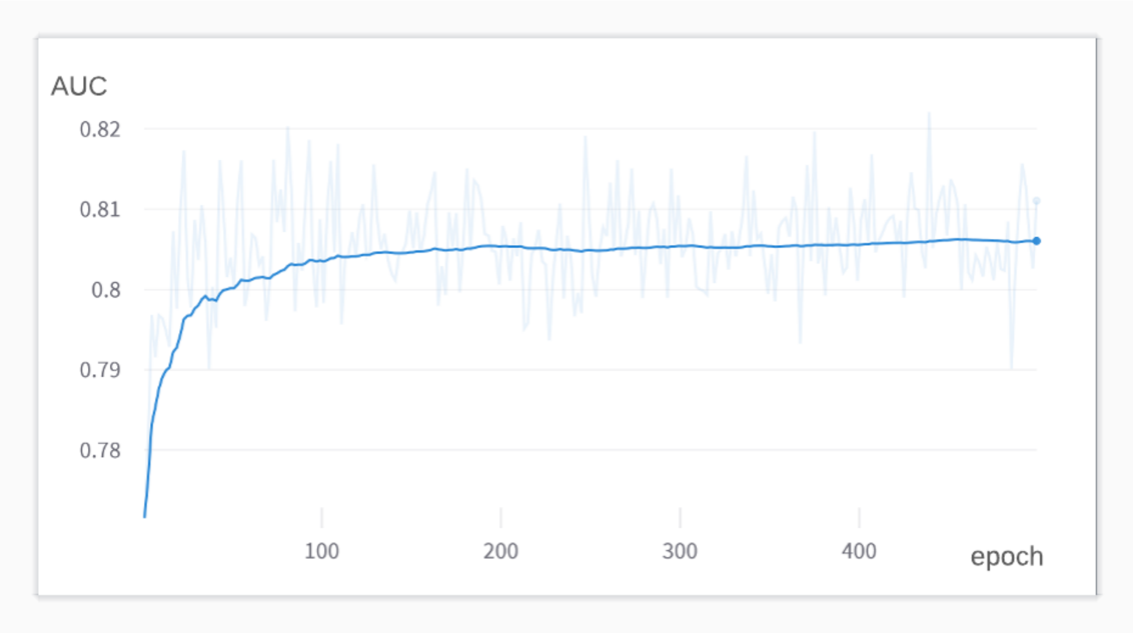


Figure 5.6: The AUC results based on I3D Flow stream network

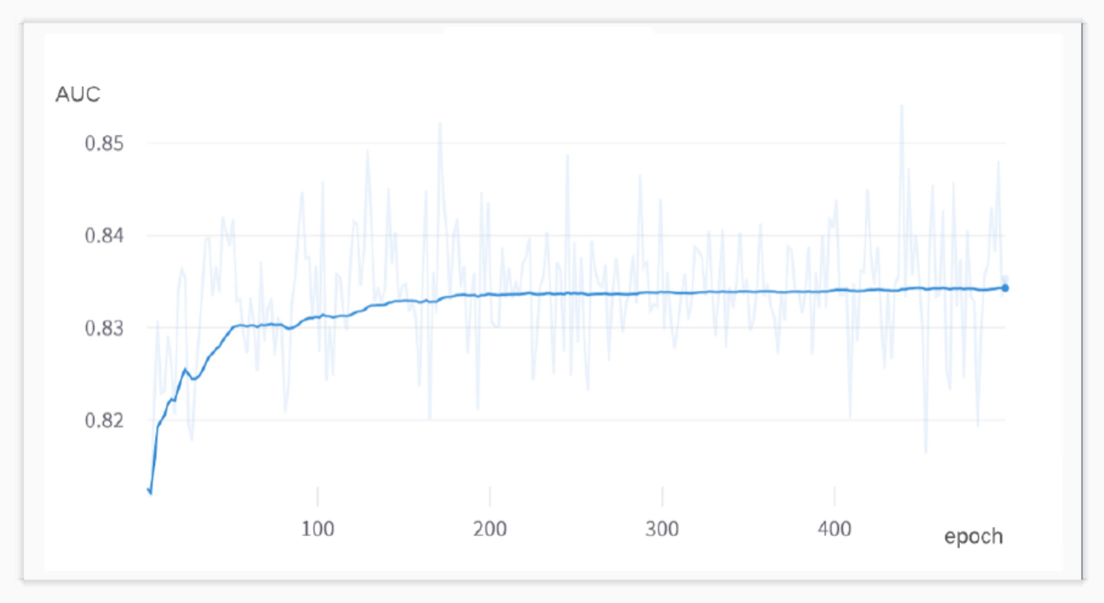


Figure 5.7: The AUC results based on two-stream I3D network

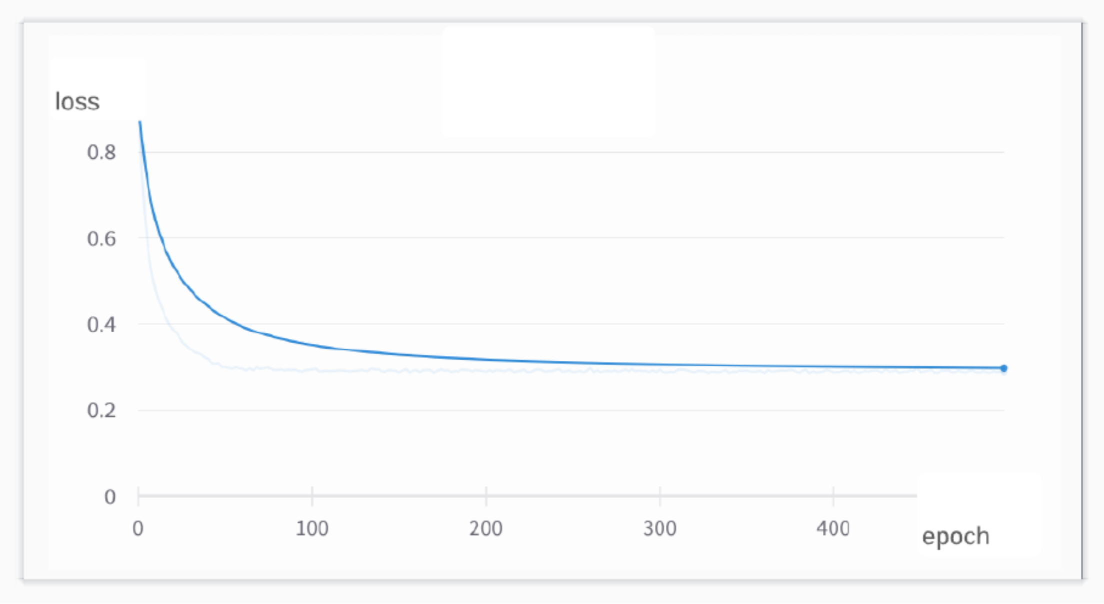


Figure 5.8: The model loss based on the two-stream network

Figure 5.9 presents a comparison of the AUC results of the three scenarios discussed in Section 5.5 on the UCF-Crime dataset. It is evident that the anomaly detection model based on the two-stream I3D network outperforms the RGB and Flow stream networks. Therefore, it can be concluded that the combination of RGB and Flow features provides more comprehensive information, which is advantageous for detecting anomalies in surveillance videos.

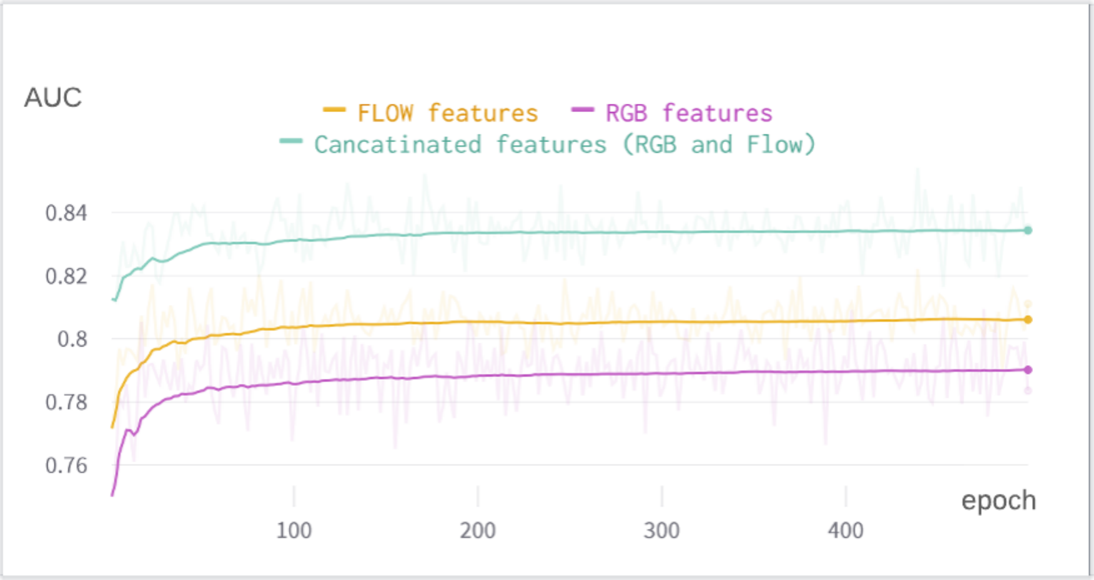


Figure 5.9: AUC results of different models on UCF-Crime

The Receiver Operating Characteristic (ROC) Curve of the proposed system is depicted in Figure 5.10, displaying the correlation between the **True Positive Rate** (TPR) and the **False Positive Rate** (FPR). This curve effectively visualizes the system's ability to differentiate between true positive detections and false positive instances.

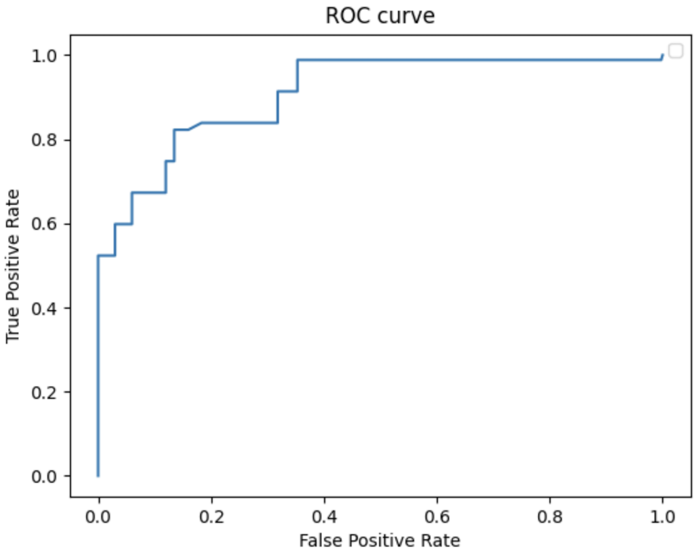


Figure 5.10: ROC Curve of the proposed method: TPR versus FPR

We compared our approach with existing anomaly detection methods. Hasan et al. [27] suggested a method based on a fully convolutional feed-forward deep autoencoder to train a classifier by learning local features. Lu et al. [47] introduced a dictionary-based method that learns normal behavior patterns and uses reconstruction errors to detect anomalies. Sultani et al. presented a model that detects anomalies by extracting C3D features. We adopted their approach as our baseline approach. Table 5.4 presents a quantitative comparison of different methods based on the AUC metric on the UCF-Crime dataset. Our proposed approach, which uses the two-stream I3D network as a feature extractor, achieves better performance than the baseline method proposed by Sultani et al. [66], as indicated by the AUC values in the table. These results demonstrate the effectiveness of our method for detecting anomalies in surveillance videos.

Method	**AUC**
Hasan et al. [27]	50.6
Lu et al. [47]	65.51
Sultani et al. (C3D) [66]	75.41
Ours (I3D RGB)	80.93
Ours (I3D Flow)	82.20
Ours (I3D RGB & Flow)	85.41

Table 5.4: Quantitative comparison on UCF-Crime

Time evaluation: The inference time of a model is a critical performance metric that measures the time taken to process an input and produce an output. In the context of anomaly detection, a fast inference time is highly desirable, as it can enable the system to quickly detect anomalous behavior and trigger appropriate responses. In our case, the measurement of approximately 7 seconds for the inference time suggests that our model is efficient in detecting anomalies in a timely manner.

5.6 Practical Application of the Proposed System

The practical implementation of the proposed system discussed in this section pertains to its potential utilization in various real-world scenarios. The anomaly detection system developed within this thesis presents opportunities for deployment in practical contexts, such as enhancing campus safety and security. Regarding campus safety, the system could be seamlessly integrated into existing surveillance systems, allowing for the monitoring and identification of abnormal activities or incidents occurring on the campus premises. Through the analysis of video streams from strategically positioned cameras across the campus, the system could effectively detect behaviors that deviate from the established norm, thereby notifying security personnel of potential threats or uncommon occurrences.

Nevertheless, it’s crucial to acknowledge that while the proposed system has showcased its effectiveness using the UCF-Crime dataset, its suitability in a distinct environment like a campus necessitates customization and fine-tuning. The definitions of normal and abnormal behavior are context-dependent, leading to the requirement for tailored adjustments. As a result, to ensure optimal performance, the proposed system would need to be retrained or fine-tuned using a dataset specifically collected from the campus setting.

Chapter 6

Conclusion

This thesis addresses the complex challenge of detecting anomalies in video surveillance. It discusses how traditional approaches to anomaly detection assume that anomalies are deviations from a learned normal pattern. However, this may not hold in surveillance videos, which capture complex real-world anomalies that cannot be constructed from normal activities. To overcome this, the thesis proposes an approach that relies less on prior information. This approach utilizes a Two-Stream Inflated 3D Convolutional Neural Network to extract both RGB and Flow features from videos. By combining both streams, a more comprehensive understanding of video content can be achieved. This improves the anomaly detection model's accuracy. The proposed method is evaluated on the UCF-Crime dataset, and the results demonstrate its superior performance compared to existing approaches to detecting anomalies in surveillance video.

6.1 Limitations and Future work

6.1.1 Limitations

Although the anomaly detection system proposed in the study achieves high performance, its computational cost is relatively high compared to simpler models such as C3D. This is due to the fact that the I3D model has more parameters and requires more computation. The I3D model utilizes two separate streams of convolutional networks to incorporate both spatial and temporal information. Each stream is specialized in processing one type of information. This means that each video frame needs to be processed twice, once for the spatial stream and once for the temporal stream. This is before the two feature sets are combined. As a result, this process requires more computation and memory compared to simpler models like C3D, which only extract temporal features. Moreover, it is important to highlight that the proposed anomaly detection method operates under a weakly supervised framework, eliminating the need for explicit annotations of anomalous events during training. While this approach brings notable advantages, such as reduced annotation effort, it can also introduce a higher risk of false positives in certain scenarios, particularly in low-light or dark scenes. To mitigate this limitation and enhance the system's accuracy, several strategies can be considered. One approach involves supplementing the training process with labeled anomalous data, thereby refining the model's understanding of diverse anomalies. Additionally, incorporating data from auxiliary sources, such as thermal imaging or audio cues, can provide complementary information that aids in distinguishing true anomalies from false positives. By carefully integrating these strategies, the overall robustness and reliability of the anomaly detection system can be substantially improved.

6.1.2 Future work

While the proposed method shows promising results, further research can be conducted in the following areas, listed below:

1. **Drone-based Anomaly Detection:** As a future direction, drones could be used to collect videos and generate a new dataset for evaluating the proposed anomaly detection model. Through this method, it could be possible to investigate the effectiveness of the model under different environmental conditions, as well as its robustness to a variety of anomalies that may not be present in the current dataset. In some scenarios, a single camera may not be enough to capture all the necessary information. In these situations, drones have several benefits over traditional cameras. As an example, drones could provide a

more comprehensive view of an area, capture footage from multiple angles and heights, as well as transmit live video to a monitoring station for constant monitoring, improving the accuracy of anomaly detection. Moreover, using drones may be more cost-effective and safer than using traditional cameras, because they can be deployed in inaccessible or hazardous areas where using humans or traditional cameras would be unsafe.

2. **Utilizing other modalities:** While the proposed approach leverages both RGB and Flow features, there are other modalities, such as depth and sound. These modalities can provide additional information about the video content. Exploring these modalities could improve anomaly detection accuracy.
3. **Dealing with occlusions:** Another limitation of the proposed approach is that it assumes that all objects in the video are visible and can be captured by the RGB and Flow features. However, in real-world scenarios, objects may be partially or fully occluded, which can lead to false positives or negatives. Future work can investigate ways to handle occlusions and improve the robustness of the anomaly detection system.
4. **Exploring the use of additional data sources:** In this thesis, we only use video data to train the anomaly detection model. Future research can explore the use of additional data sources such as audio, text, and sensor data to improve the model's accuracy.
5. **Investigate the integration of human-in-the-loop approaches**: Human operators can enhance the performance of anomaly detection system by providing feedback.

Bibliography

[1] Example of a convolutional neural network.png. `https://en.wikipedia.org/wiki/Convolutional_neural_network#/media/File:Typical_cnn.png`.

[2] Example of a deep neural network.png. `https://upload.wikimedia.org/wikipedia/commons/2/2f/Example_of_a_deep_neural_network.png`.

[3] How to interpret a roc curve. `https://www.statology.org/wp-content/uploads/2021/08/read_roc1-768x598.png`.

[4] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. pages 481–490, 06 2019.

[5] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, 2008.

[6] KP Adhiya, SR Kolhe, and Sandip S Patil. Tracking and identification of suspicious and abnormal behaviors using supervised machine learning technique. In *Proceedings of the International Conference on Advances in Computing, Communication and Control*, pages 96–99, 2009.

[7] Samet Akcay, Amir Atapour Abarghouei, and Toby Breckon. *GANomaly: Semi-supervised Anomaly Detection via Adversarial Training*, pages 622–637. 05 2019.

[8] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.

[9] Borislav Antic and Björn Ommer. Video parsing for abnormality detection. *2011 International Conference on Computer Vision*, pages 2415–2422, 2011.

[10] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[11] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, 41(3):15, 2009.

[13] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2909–2917, 2015.

[14] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. pages 189–196, 01 2017.

[15] Xinyi Cui, Qingshan Liu, Mingchen Gao, and Dimitris Metaxas. Abnormal detection using interaction energy potentials. pages 3161–3167, 06 2011.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[17] John Duchi and Elad Hazan. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 07 2011.

[18] Hongxiang Fan, Ho-Cheung Ng, Shuanglong Liu, Zhiqiang Que, Xinyu Niu, and Wayne W. C. Luk. Reconfigurable acceleration of 3d-cnns for human action recognition with block floating-point representation. *2018 28th International Conference on Field Programmable Logic and Applications (FPL)*, pages 287–2877, 2018.

[19] Max Ferguson, Ronay ak, Yung-Tsun Lee, and Kincho Law. Automatic localization of casting defects with convolutional neural networks. pages 1726–1735, 12 2017.

[20] Damien Fourure, Muhammad Javaid, Nicolas Posocco, and Simon Tihon. *Anomaly Detection: How to Artificially Increase Your F1-Score with a Biased Evaluation Protocol*, pages 3–18. 09 2021.

[21] Dong Gong, Lingqiao Liu, Lê Vng, Budhaditya Saha, Moussa Mansour, Svetha Venkatesh, and Anton Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, 04 2019.

[22] Dong Gong, Lingqiao Liu, Lê Vng, Budhaditya Saha, Moussa Mansour, Svetha Venkatesh, and Anton Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, 04 2019.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014.

[24] Kasthurirangan Gopalakrishnan, Hoda Gholami, Akash Vidyadharan, Alok Choudhary, and Ankit Agrawal. Crack damage detection in unmanned aerial vehicle images of civil infrastructure using pre-trained deep learning model. *International Journal for Traffic and Transport Engineering (IJTTE)*, 8:1–14, 02 2018.

[25] Wangli Hao, Ruixian Zhang, Shancang Li, Junyu Li, Fuzhong Li, Shanshan Zhao, and Wuping Zhang. Anomaly event detection in security surveillance using two-stream based model. *Security and Communication Networks*, 2020, 2020.

[26] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit Roy-Chowdhury, and Larry Davis. Learning temporal regularity in video sequences. 04 2016.

[27] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016.

[28] Chengkun He, Jie Shao, and Jiayu Sun. An anomaly-introduced learning method for abnormal event detection. *Multimedia Tools and Applications*, 77:29573–29588, 2018.

[29] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[30] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A markov clustering topic model for mining behaviour in video. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1165–1172. IEEE, 2009.

[31] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016.

[32] Lakhmi C. Jain and Larry R. Medsker. Recurrent neural networks: Design and applications. 1999.

[33] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[34] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[35] Julian Kooij, Martijn Liem, Dirkjan Krijnders, T.C. Andringa, and Dariu Gavrila. Multi-modal human aggression detection. *Computer Vision and Image Understanding*, 144, 01 2015.

[36] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. pages 1446–1453, 06 2009.

[37] Louis Kratz and Ko Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1446–1453. IEEE, 2009.

[38] Jayant Kumar, Jaishanker Pillai, and David Doermann. Document image classification and labeling using multiple instance learning. pages 1059 – 1063, 10 2011.

[39] Joo-Yeon Lee, Woojeoung Nam, and Seong-Whan Lee. Multi-contextual predictions with vision transformer for video anomaly detection, 06 2022.

[40] Nannan Li, Xinyu Wu, Huiwen Guo, Dan Xu, Yongsheng Ou, and Yen-Lun Chen. Anomaly detection in video surveillance via gaussian process. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(06):1555011, 2015.

[41] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.

[42] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):18–32, 2014.

[43] Wen Liu, Weixin Luo, Zhengxin Li, Peilin Zhao, Shenghua Gao, et al. Margin learning embedded prediction for video anomaly detection with a few anomalies. In *IJCAI*, pages 3023–3030, 2019.

[44] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection - a new baseline. pages 6536–6545, 06 2018.

[45] Yeqi Liu, Huihui Yu, Chuanyang Gong, and Yingyi Chen. A real time expert system for anomaly detection of aerators based on computer vision and surveillance cameras. *Journal of Visual Communication and Image Representation*, 68:102767, 2020.

[46] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[47] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[48] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, 2020.

[49] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. pages 341–349, 10 2017.

[50] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997.

[51] Jefferson Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. 12 2016.

[52] Jefferson Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. 12 2016.

[53] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. In *2009 IEEE conference on computer vision and pattern recognition*, pages 935–942. IEEE, 2009.

[54] Tianhui Meng, Katinka Wolter, Huaming Wu, and Qiushi Wang. A secure and cost-efficient offloading policy for mobile cloud computing against timing attacks. *Pervasive and Mobile Computing*, 45:4–18, 2018.

[55] Sadegh Mohammadi, Alessandro Perina, Hamed Kiani, and Vittorio Murino. Angry crowds: Detecting violent events in videos. volume 9911, pages 3–18, 10 2016.

[56] Vandana Mohindru and Shafali Singla. *A Review of Anomaly Detection Techniques Using Computer Vision*, pages 669–677. 01 2021.

[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[58] Yujiang Pu and Xiaoyu Wu. Locality-aware attention network with discriminative dynamics learning for weakly supervised anomaly detection. *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2022.

[59] Mohammad Sabokrou, Mahmood Fathy, and H. Mojtaba. Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder. *Electronics Letters*, 52, 04 2016.

[60] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 02 2018.

[61] Imran Saleemi, Khurram Shafique, and Mubarak Shah. Probabilistic modeling of scene dynamics for applications in visual surveillance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31:1472 – 1485, 09 2009.

[62] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[63] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[64] Virender Singh, Swati Singh, and Pooja Gupta. Real-time anomaly recognition through cctv using neural networks. *Procedia Computer Science*, 173:254–263, 01 2020.

[65] Sandeep Sony, Kyle Dunphy, Ayan Sadhu, and Miriam Capretz. A systematic review of convolutional neural network-based structural condition assessment techniques. *Engineering Structures*, 226:111347, 01 2021.

[66] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. 01 2018.

[67] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[68] Yu Tian, Guansong Pang, Yuanhong Chen, Rajvinder Singh, Johan W Verjans, and Gustavo Carneiro. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4975–4986, 2021.

[69] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[70] Adarsh Trivedi, Siddhant Srivastava, Apoorva Mishra, A. Shukla, and Ritu Tiwari. Hybrid evolutionary approach for devanagari handwritten numeral recognition using convolutional neural network. *Procedia Computer Science*, 125:525–532, 01 2018.

[71] Boyang Wan, Yuming Fang, Xue Xia, and Jiajie Mei. Weakly supervised video anomaly detection via center-guided discriminative learning. *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020.

[72] Ce Wang, Zhangling Chen, Kun Shang, and Huaming Wu. Label-removed generative adversarial networks incorporating with k-means. *Neurocomputing*, 361:126–136, 2019.

[73] Shandong Wu, Brian Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. pages 2054–2060, 06 2010.

[74] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for real-time tv-l1 optical flow. In *DAGM-Symposium*, 2007.

[75] M. Zaigham Zaheer, Jin Ha Lee, Seung-Ik Lee, and Beom-Su Seo. A brief survey on contemporary methods for anomaly detection in videos. In *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 472–473, 2019.

[76] Jiangong Zhang, Laiyun Qing, and Jun Miao. Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection. *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4030–4034, 2019.

[77] Kaifeng Zhang, Dan Li, Jiayun Huang, and Yifei Chen. Automated video behavior recognition of pigs using two-stream convolutional networks. *Sensors*, 20(4), 2020.

[78] Mingyang Zhang, Tong Li, Yue Yu, Yong Li, Pan Hui, and Yu Zheng. Urban anomaly analytics: Description, detection, and prediction. *IEEE Transactions on Big Data*, 8(3):809–826, 2022.

[79] Qianyun Zhang, Kaveh Barri, Saeed Babanajad, and Amir Alavi. Real-time detection of cracks on concrete bridge decks using deep learning in the frequency domain. *Engineering*, 7, 11 2020.

[80] Bin Zhao, Li Fei-Fei, and Eric P. Xing. Online detection of unusual events in videos via dynamic sparse coding. *CVPR 2011*, pages 3313–3320, 2011.

[81] Yuxuan Zhao. *Deep Learning in Video Anomaly Detection and Its Applications*. PhD thesis, University of Liverpool, 2021.

[82] Yuxuan Zhao, Ka Man, Jeremy Smith, Kamran Siddique, and Sheng-Uei Guan. Improved two-stream model for human action recognition. *EURASIP Journal on Image and Video Processing*, 2020, 06 2020.

[83] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H. Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. *CoRR*, abs/1903.07256, 2019.

[84] Yuanyi Zhong, Jianfeng Wang, Lijuan Wang, Jian Peng, Yu-Xiong Wang, and Lei Zhang. Dap: Detection-aware pre-training with weak supervision. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4535–4544, 2021.

[85] Yuanyi Zhong, Jianfeng Wang, Lijuan Wang, Jian Peng, Yu-Xiong Wang, and Lei Zhang. Dap: Detection-aware pre-training with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4537–4546, June 2021.

[86] Joey Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Goh. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, PP:1–1, 02 2019.

[87] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 08 2017.

[88] Yingying Zhu, Nandita Nayak, and Amit Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *IEEE Journal of Selected Topics in Signal Processing*, 7:91–101, 02 2013.

[89] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2017.

Curriculum Vitae

Name:	Sareh Soltani Nejad
Post-Secondary	M.Sc. Candidate, Computer Science (2021-2023) The University of Western Ontario
Awards:	Western Graduate Research Scholarship (WGRS)
Experience	Teaching Assistant University of Western Ontario 2021-2023 Teaching Assistant Amirkabir University of Technology 2014-2019