Anomaly detection in videos

by

SUYASH DHONDKAR
202011035

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

July, 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Dhondkar

Suyash Dhondkar

Certificate

This is to certify that the thesis work entitled **"Anomaly detection in videos"** has been carried out by **Suyash Dhondkar** (202011035) for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.

M Khare 14/7/22

Prof. Manish Khare
Thesis Supervisor

Prof. Pankaj Kumar
Thesis Supervisor

Acknowledgments

First and foremost, I would want to convey my heartfelt gratitude to my supervisors, Prof. Manish Khare, and Prof. Pankaj Kumar, for their unwavering and invaluable professional mentoring during the last year. I appreciate their patience during our weekly/monthly sessions, especially given that we were meeting virtually for the bulk of the time due to the ongoing COVID-19 outbreak. Their advice and constant motivation was vital in my research.

I would also like to thank Prof.Avik Hati for teaching the course on Computer Vision. The knowledge and insights I gained in that course helped me lot in understanding the subject which was essential for my research.

I express my gratitude to my alma mater Dhirubhai Ambani Institute of Information and Communication Technology for giving me the opportunity to pursue my masters in their esteemed institute. Special thanks to the the entire team of DAIICT Help Desk for ensuring that I was able to access the institute resources remotely.

Finally I would like to thank my family for their constant support and encouragement throughout the last year.

Contents

Abstract

With the development of modern digital cameras and CCTVs, our cities and their important public places can now be monitored 24x7 around the clock. Traditional video analytics methods necessitate ongoing surveillance monitoring, which is time-consuming. A lot of human resource is needed for running such systems to observed and react to any abnormal event. Hence there is a need to develop automatic video anomaly detection systems to reduce the dependency on human resources. An anomaly can be described as an extraordinary event or an emergency that differs from the norm. Finding and classifying anomalies in videos is known as video anomaly detection. This thesis discusses the various algorithms and techniques that are used to create anomaly detection systems. We have proposed a model using Multiple Instance deep learning network for solving the problem of video anomaly detection. We have used the two-stream Inflated 3-D Convolutional Neural Network (I3D) for feature extraction from the RGB and optical flow data stream. We propose a modified loss function based on the deep ranking loss criteria to improve the model's effectiveness. For training and testing the model, we have used the UCF-Crime dataset. To check the model's effectiveness, we have used the Area Under Curve (AUC) value of the Receiver Operating Characteristic (ROC) curve and compared it with the state of the art methods We have also compared the loss resolution of the standard ranking loss function with that of our modified loss function. Finally we have compared the anomaly activity classification accuracy of our proposed model with that of the state of art models.

List of Tables

List of Figures

Chapter 1
Introduction

In the past few decades, we have seen an increase in the number of crimes ranging from acts of vandalizing public property by a mob to terror incidents. Our law enforcement agencies carry out video surveillance to avoid escalation of such events and provide a rapid response to quell them. Technological development has also led to a significant improvement in the quality of the videos collected, resulting in a significant increase in the volume of information stored. There is, therefore, a need to process a large amount of information in the form of video files. Historically, such processing has been done manually, consuming many human resources. Today, the speed at which this information is generated, and the sheer volume of information generated daily, make it almost impossible to manage this information manually in a thorough, exhaustive, and adequate manner. This information must be processed in real-time as much as possible since rapid response in emergencies is crucial to reduce the effects of a potential catastrophe. Thus, the need for automatic video monitoring systems has arisen to analyze video footage for anomalous activities and incidents.

Anomaly detection is described as the process of finding outliers in a set of data points that do not conform to the significant cluster of the dataset. Such points deviate from expected values. An anomaly can be an abrupt change in the values when historical data might show the values to be constant or converse. It is difficult to judge whether a given deviation is good or bad without having any context based on previous data [1]. Thus it is necessary to create comprehensive datasets to create a model for what can be termed as “normal” behavior. Another important aspect is the need for a quantifiable metric to decide the degree of anomaly, an anomaly score. We now describe the idea behind creating video anomaly detection systems and the motivation for using deep learning.

Video anomaly detection [1] is a branch of research whose objective is the analysis of multiple real-time video sources for the automatic extraction of relevant information related to the behavior of individuals and to notify the presence of abnormal behavior patterns. This research area brings together two necessary fields of work within the machine learning/deep learning domains; computer vision and time series analysis. Since the most common type of data in this context is video sequences, information has to be extracted from each frame. The information extracted from these individual frames is then analyzed over the temporal domain to get more context from them.

Video anomaly detection systems focus on the following tasks of video analysis :

1. Detection and tracking of individuals
2. Counting and density estimation of individuals
3. Behavior analysis and classification
4. Detection of anomalous behaviors

Tasks 1 and 2 have been extensively studied, and classical learning models are found to provide sufficiently good results. However, when it comes to tasks 3 and 4, the traditional machine learning methods are ineffective. Today, the volume of video information has increased, along with our computational capacity. Hence we can use deep learning techniques as they thrive on a large volume of data. In this thesis, we predominantly focus on the problem of anomalous behavior in video sequences by analyzing the images or image segments using deep learning techniques. We now describe the various characteristics of anomalies found in surveillance videos.

1.1 Characteristics video anomalies

The anomalies that are found in surveillance videos are abnormal behavior patterns of the general public [1] depending upon the location. The anomalies in video footage [2] from places like a traffic stop, banks, and house backyards can potentially alert for the occurrence of a violent crime, accidents, robberies, etc. On the other hand, the abnormal incidents at public places like banks and traffic signals would be different. Hence, we need to analyze video information from diverse environments to learn anomalous behavior patterns. Figure 1.1 shows some anomalies we encounter in the public places like road accidents, armed robbery, vandalizing public property, public shooting. When it comes to analyzing the behavior of individuals, we limit ourselves to human-human and human-environment interactions [3]. When we consider anomalies in public places, the density of crowds surrounding the location plays an important role; hence, crowd density is essential in such cases. In a place like a crowded railway station, anomalies like a stampede or panic dispersion situation are quite different from two people fighting or a thief trying to steal. In a surveillance video, the anomaly can occur in a fixed time slice, and the rest of the video can be "normal", also in a given clip (set of continuous frames), the anomaly can occur only in a localized spot, and the rest of frame might be unchanged.

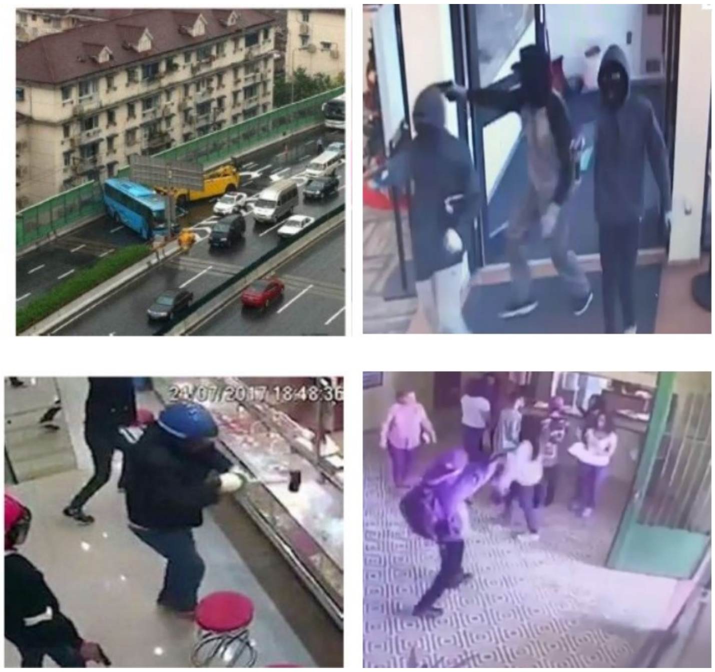


Figure 1.1: Various types of abnormal events seen in cities.

1.2 Motivation

The problem of finding anomalies in videos is critical in smart city management, such as traffic control and criminal investigation. Unlike other anomaly detection tasks that can yield clear unusual signals, video anomaly identification necessitates video analysis. For carrying out such analysis manually around the clock and everyday, many human resources would be utilized. Hence there is a need for automatic video anomaly detection systems. Research in automated video anomaly detection systems is of great practical importance as it can help reduce the large amount of human effort needed to carry out surveillance. Furthermore, it can help develop better systems to identify an anomalous event and send an alert to resolve such incidents.

1.3 Aims and objectives

The primary aim of this thesis is to construct a video anomaly detection system based on the latest deep learning techniques. We aim to use a two-stream feature extraction network to provide a composite feature vector to train our deep learning classifier. Another aspect we focus on is modifying the standard loss function to improve the accuracy and reduce the training epochs of the classifier. As we are dealing with videos, the aim is to extract more contextual information from multiple anomalous video segments, thus increasing the size of the temporal dimension resulting a more effective classification model.

1.4 Organization of the thesis

The thesis henceforth is organized as follows:

In chapter 2, we provide the literature survey. We provide a brief history of this problem, and the initial approaches proposed to tackle this problem. In chapter 3, we propose our method for creating a video anomaly detection system. Chapter 4 provides the details of the datasets we use for training and testing the models. We analyze the performance of our model on various parameters with the state of the art methods. Chapter 5 concludes the thesis and provides potential pathways for future work in this domain.

CHAPTER 2

Literature Review

This chapter describes the three major strategies, including traditional image analysis techniques like object tracking, orientation and motion of people, and their interactions. Then we look at the classical machine learning techniques like supervised learning and non-supervised clustering. Finally, we describe the deep learning architectures used for creating anomaly detection systems.

2.1 Handcrafted feature based video anomaly detection

Traditional methods of video anomaly detection rely on low-level features extracted from consecutive frames. They rely on annotations of every frame to find the area of interest in the entire visual scene. An area of interest can be a person or object, and their location or orientation changes. Fernyhough et al. [4] proposed one of the earliest methods to find such areas in images called semantic regions. They proposed a 2-D tracking technique that used surveillance footage to create paths for the entities in the scene. These trajectories are then compared to detect irregular patterns in the motion of obejcts or people.

2.1.1 Trajectory based approaches

The concept behind clustering-based approaches is that an anomaly is usually unexpected and appears in a wide variety of videos. As a result, these algorithms can learn regular trajectories from ordinary video occurrences. Tung et al. [5] present a goal-based framework based on a mix of three models. A spatial scene model is learned in the training phase; a region transition model is trained to incorporate movement statistics between spatial regions, and trajectories in progress are identified using particle filtering in a probabilistic framework. These models perform a trajectory analysis for abnormal patterns in the paths followed by people and provide a probability of the presence of an anomaly. Calderara et al. [6] have proposed a trajectory analysis method by using spectral graphs. They have created a representation of the trajectory paths by randomly selecting points on them using Voronoi tessellation. A Voronoi tessellation divides a plane into zones that are near to each of a set of objects. These objects can be a finite number of points in the plane called seeds, sites, or generators. There is a Voronoi cell for each seed, consisting of all points in the plane closer to that seed than any other. Transitions between neighboring Voronoi cells are represented as vertices in the graph, and a set of two successive transitions are represented as weights on the graph edges. The movement of people in various Voronoi cells is then analyzed using the spectral graphs to find anomalous patterns.

2.1.2 Low-level feature extraction

Trajectory-based methods are overly reliant on tracking the entities' motion and their location in the frame. Furthermore, these methods are complex and challenging to use in clustering techniques. To overcome this issue, low-level feature extraction techniques are used. Low-level feature extraction approaches concentrate on low-level video displays such as greyscale shifts, moving flow vectors, and textures. Adam et al. [7] have provided a multi-camera framework to monitor optical flow vectors from various orientations. They have aggregated the flow vectors from multiple cameras to detect alarming frames. Wang et al. [8] have proposed a texture-based video anomaly localization and detection technique. They have given a novel Robust Principal Component Analysis (PCA) video foreground localization technique to locate the anomaly areas. They combined the Light Gradient Patterns (LGP) for textures and optical flow vectors and proposed a one-class classification method.

2.2 High-level feature based video anomaly detection methods

In image analysis, high-level features are built using low-level features like texture, flow vectors, Scale Invariant Feature Transform(SIFT), etc. High-level features are layered on top of low-level features to detect objects and larger shapes in the image. These features have been used to create machine learning models based on clustering methods. The anomaly detection approach of Lu et al. [9] was based on computing gradient-based features on high frame rate videos. They utilized a dictionary-based system to learn normal activity features and used them to create a model. Hasan et al. [10] have implemented an autoencoder-based approach to detect temporal irregularities in consecutive video frames. Yang et al. [2] have used spatio-temporal features to train an autoencoder to detect abnormal actions in large crowds.

In [11] they have used a Gaussian Mixture Model (GMM) to create a clustering model using the feature vectors. A new framework for detecting anomalies was introduced in [11] for weakly or partially labeled datasets. The datasets used for video anomaly detection are usually unlabelled; hence clustering algorithms are prevalent for creating models. However, clustering techniques are ineffective in detecting abnormal behavior when the changes are not very distinct. These models are prone to high false alarm rates [1] and are not very effective in creating a generalized solution [12]. Liu et al. [13] have proposed a Generative Adversarial Networks(GANs) model to detect outliers in nonlabelled datasets. The advantage of using GANs is that they can adapt to any unseen change by updating the model from real-world data.

2.3 Deep learning based video anomaly detection

A deep learning-based video anomaly detection system combines a feature extractor and a deep learning classifier. Feature extraction is converting the information from the raw or original form into a vector space. This involves converting an image or a sequence of images into an N-dimensional vector based on contextual features in the computer vision domain. The main task of the classifier is to use the feature vector to identify the presence of an anomaly in the video. A barebones video anomaly detection system is illustrated in Figure 2.1. The video is divided into fixed frame count snippets and then passed through a 3D Convolutional Neural Network (CNN). The CNN extracts contextual spatio-temporal features, and then the feature vectors are used to train the classifier. We will now look at the CNN-based feature extraction networks and the various learning techniques to train the classifier.

2.3.1 Feature extraction with 3D CNN

CNN is a deep learning technique for analyzing data by applying a series of convolutions one after another, which reduce the dimensionality of the input space. A convolution is a function that expresses the amount of overlap of one function

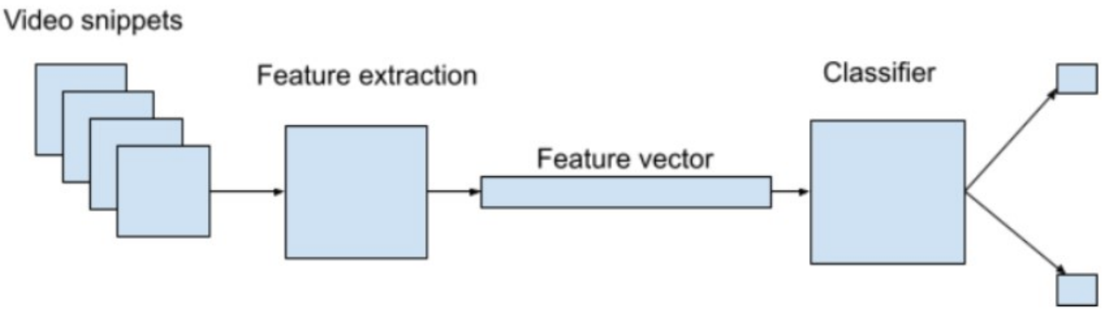


Figure 2.1: A bare-bones architecture of deep learning based video anomaly detection model.

as it is shifted over another function. It "blends" one function with another. Next, a pooling layer also reduces data dimensionality by taking either an average or a max value from a block of convoluted values. These are then passed on to the next layer of convolutions and poolings until the desired result is achieved. Unlike other neural networks, CNNs have replaced the matrix multiplication layers with the convolution layers. As a result of this change, the computation complexity is reduced, leading to lesser resource consumption. Furthermore, due to the nature of their architecture, individual images can be directly passed into the networks as input. There is no need for any traditional feature extraction or dimensionality reduction. CNNs are capable of extracting both low-level and high-level features and provide a combined result. For all these features, CNN has been the most successful and popular NN architecture, especially for image analytics. We require 3D CNN networks to factor in the extra temporal dimension for feature extraction from videos. We now describe the popular 3D CNN for extracting spatio-temporal features from consecutive video frames along with the alogorithm for extracting optical features. We also describe the weakly supervised deep learning techniques and also provide a justification for using them.

The 3D Convolutional Neural Network (C3D)

One of the most widely used CNN models is the C3D model developed on the UCF-101, and Sports-1M action recognition datasets by Tran et al.[14]. Action recognition is the task of classifying the actions performed in videos. C3D is a deep 3-dimensional convolutional neural network with a homogenous architecture containing convolutional kernels followed by max-pooling at each layer. 2D ConvNets like Alexnet or ResNet focus only on the spatiometric changes and average the temporal information across frames. The architecture of the C3D network is shown in Figure 2.2. Hence they are not particularly effective at action recognition. The 3D convolutions extract features relating to the motion of objects, human actions, human-scene or human-object interaction, and the appearance of those objects, humans, and scenes. The deeply interconnected nature of C3D allows it to pick up changes in the spatiometric features along the temporal dimension. This makes sure that C3D filters selectively focus on appearance and motion at different instants of a video segment.

The first layer of the C3D network is a 1x3x3 convolution layer, followed by a 1x2x2 pooling layer. This is done so that the temporal information is preserved in the first layer, and higher-level representations of the temporal information can be built in the network's following levels. Each subsequent convolution and pooling layer would be 3x3x3 and 2x2x2, respectively, with strides of 1 and 2. The activation function at each layer is ReLu (Rectified Linear activation $f(x) = max(0, x)$), the final fully connected layer has a sigmoid activation function. The final fully connected layer (fc7 in Figure 2.2) provides a 4096-D feature vector. The sigmoid function flattens this into a softmax out, reflecting 101 classes from the UCF-101 dataset or 487 classes from the Sports 1M dataset. For anomaly detection, we do not pass the output of the final fully connected layer through the softmax classification; rather, we use the output to train the classifier.

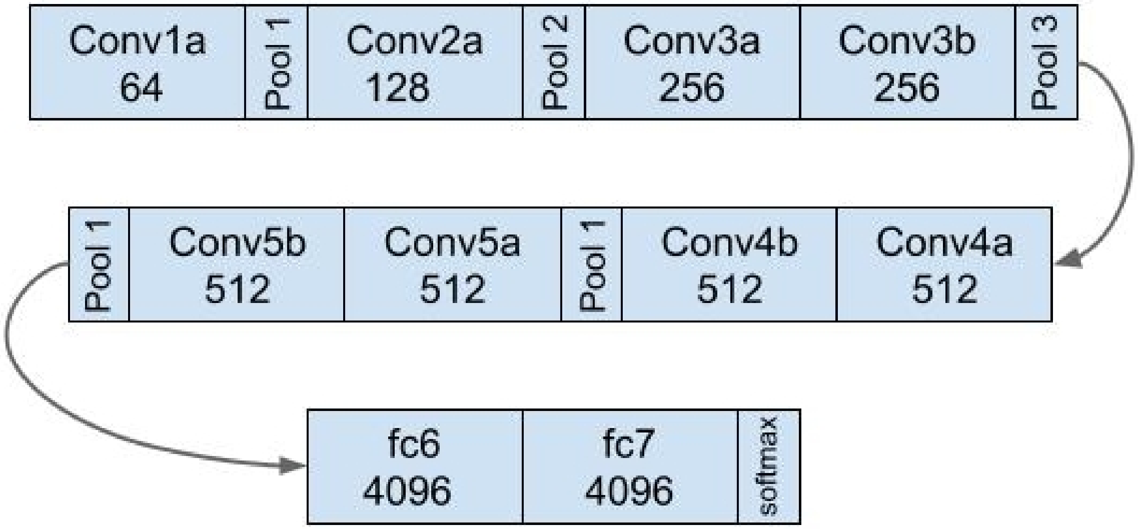


Figure 2.2: C3D network architecture.

Inflated 3-D Convolutional Neural Network (I3D)

The temporal element is one of the critical differences between information in a single image and information in a video. As a result, deep learning model architectures have been inflated and improved to include 3D processing with temporal data. I3D is one such 3D ConvNet introduced by DeepMind and the University of Oxford researchers in [15]. This network was trained on the Kinetics Human Action Video Dataset [16], which has around 400 human actions categorized in it. The I3D network is initialized with a 2D architecture and inflated by the numerous convolution filters and pooling kernels. The inflating process adds the temporal dimension, thus making it a 3D capable network. The initial layer of the I3D net is bootstrapped with the parameters of a 2D dense CNN trained on a large dataset like ImageNet. The same parameters are then applied to N such images to form a snippet. A layer of max and average pooling aggregates the result of multiple frames. Thus, it works like a dense CNN applied on consecutive frames aggregating the results for each one.

Another feature of the I3D network is its large receptive field of pooling and convolution layers. A receptive field for a CNN is defined as the part of the image that a filter can process at one time. The size of the receptive field increases as the number of layers in the network increase. 2D convolutions and pooling are symmetrical since they focus on the image's height and width. However, the appropriate receptive field must be determined when a temporal dimension is added, dependent on the frame rate and image size. Suppose the receptive field increases too quickly in time compared to space. In that case, it may mix edges from various objects, breaking early feature identification, as pointed out by Carreira et al. [15]. The receptive field may not capture scene dynamics as well if it increases too slowly. Because of the added time dimension, the kernels in I3D are not symmetrical, and layers are deeper than in other action recognition networks like C3D.

The architecture of the I3D network is shown in Figure 2.3 with all the layers and the Inception-V1 module. The stack on N images is input into the first 7x7x7 convolution layer with the weights of a dense CNN architecture. Subsequently, the network applies max-pooling to increase the receptive field, as shown in the diagram. The receptive field is represented as TxWxH, where T denotes the number of frames (size of temporal dimension), and WxH denotes the size of the image being processed at a time. The final size of the receptive field is 99x539x539 which is large enough to cover any substantial part of a video. The Inception-V1 module significantly increases the size of the receptive field in the spatial dimension, making it wider. The advantage of using a Inception module is that it greatly reduces the number of computations needed to perform a convolution by applying smaller convolutions repeatedly than applying one large at a time. The final fully

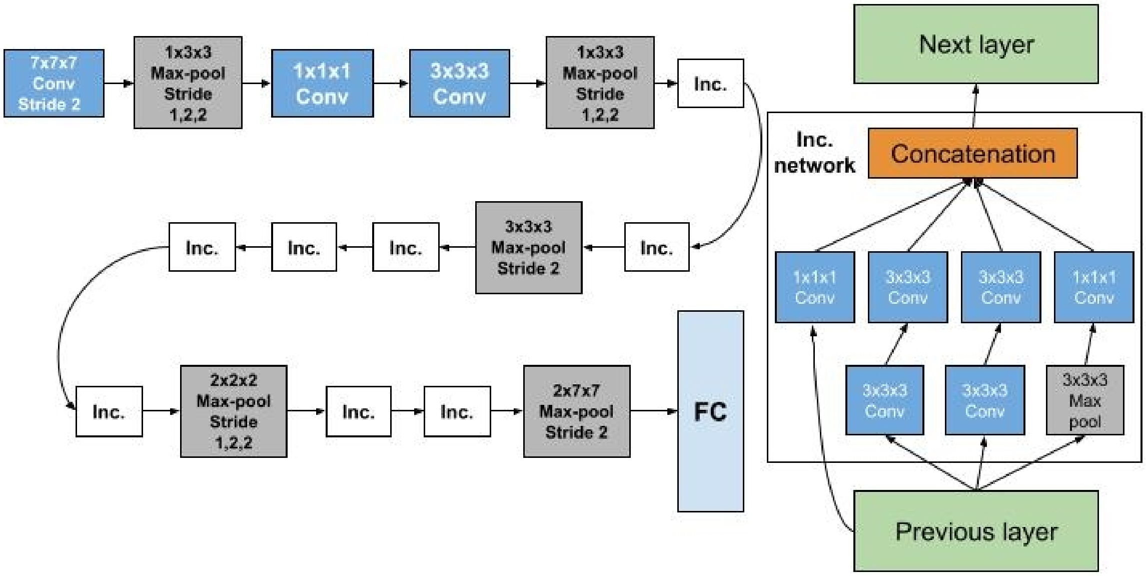


Figure 2.3: The architecture of I3D network along with the Inception-V1 module [15]

connected layer applies a sigmoid activation function to output a 1024D dimension vector which can be used to train the classifier.

2.3.2 Optical flow and the TV-L1 algorithm

We have seen various CNNs for extracting spatio-temporal RGB features from video frames. However, when it comes to analyzing videos, tracking the motion of various entities in the frame is essential. The pattern of apparent motion of objects, surfaces, and edges in a visual scene induced by the relative motion between an observer and a scene is known as optical flow. The distribution of apparent velocities of movement of a brightness pattern in an image is also known as optical flow.

Mathematically, optical flow is defined for every pixel (x, y) at time t as the change in the value of the intensity $I(x, y, t)$ of its over a time interval Δt between two consecutive frames as follows ,

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{2.1}$$

After applying Taylor series expansion and truncating the higher order terms,

$$\frac{\delta I}{\delta x}\Delta x + \frac{\delta I}{\delta y}\Delta y + \frac{\delta I}{\delta t}\Delta t = 0 \tag{2.2}$$

after dividing by Δt let $\frac{\Delta x}{\Delta t} = V_x$, $\frac{\Delta y}{\Delta t} = V_y$ the expression becomes,

$$\frac{\delta I}{\delta x} V_x + \frac{\delta I}{\delta y} V_y + \frac{\delta I}{\delta t} = 0 \tag{2.3}$$

This is the Optical flow equation, here V_x, V_y represent the velocity or flow in the x and y direction respectively, taking the $arctan(V_y/V_x)$ gives the direction of flow. $\frac{\delta I}{\delta x}, \frac{\delta I}{\delta y}, \frac{\delta I}{\delta t}$ are the partial derivatives of $I(x, y, t)$. There are two unknowns in this equation V_x and V_y, to solve the equation various methods have been proposed based on certain assumptions. The TV-L1 algorithm is based on the Horn–Schunck method of optical flow estimation. Equation 2.3 can be written in vector operator form as follows with vector field $u(x, y) = (u1(x, y), u2(x, y))$

$$\nabla I.u + \frac{\delta I}{\delta x} = 0 \tag{2.4}$$

The TV-L1 algorithm

The optical flow constraint given by Equation 2.4 is an underdetermined linear system. The addition of a smoothness requirement, which somehow causes u to be regular, is a common technique to solve underdetermined systems. Horn and Schunck proposed choosing the u that minimizes the following function:

$$\int_{\Omega} \left(\nabla I.u + \frac{\partial I}{\partial t} \right)^2 + \alpha \left(|\nabla u_1|^2 + |\nabla u_2|^2 \right) \tag{2.5}$$

Standard methods may easily handle this minimization problem, and the resulting flow estimates are adequate for many tasks. The $|\nabla_1|^2 + |\nabla u_2|^2$ term's fundamental flaw penalizes high u gradients and essentially eliminates discontinuities. If the image data is continuous in time, Equation 2.4 is appropriate. To accommodate for general image sequences, this equation is typically substituted by the non-linear formulation $I_1(x + u) - I_o x = 0$. Using Taylor expansions, the non-linear term $I_1(x + u)$ can be linearized, yielding the equation below,

$$\nabla I_1 \left(x + u^0 \right) .(u - u^0) + I_1(x + u^0) - I_0(x) = 0 \tag{2.6}$$

here u^0 is a close approximation of u. By modifying the quadratic components in the Horn–Schunck functional, it is possible to enable discontinuities in the flow field, resulting in the approach detailed here. The TV-L1 algorithm minimizes the following function, which is the sum of the overall fluctuation of u and an $L1$

attachment term.

$$\int_{\Omega} |\nabla u_1| + |\nabla u_2| + \lambda \, |\varrho(u)| \tag{2.7}$$

Sanchez et al. [17] have used the convex relaxation approach to minimize this function. The final equation $E_\theta(u,v)$ is,

$$E_\theta(u,v) = \int_{\Omega} |\nabla u_1| + |\nabla u_2| + \frac{1}{2\theta} \, |u-v|^2 + \lambda \, |\varrho(v)| \tag{2.8}$$

When $u$and v are approximately equal, the minimum of E occurs, reducing to the original value of E, described in Equation 2.7. This relaxation is interesting because E can be minimised by fixing one of u or v and solving for the other variable. The Equation 2.8 provides an approximate value of optical value, Sanchez et al.[17] have proposed a solution of this equation in their work known as TV-L1 algorithm. When $u$and v are approximately equal, the minimum of E occurs, reducing to the original value of E, described in Equation 2.7. This relaxation is interesting because E can be minimized by fixing one of uor v and solving for the other variable. To solve Equation 2.8, the algorithm proposes a two-step approach,

1. Fix v and find the minimum value of u,

$$min_u \int_{\Omega} |\nabla u_1| + |\nabla u_2| + 1/2\theta \, |u-v|^2 \tag{2.9}$$

2. Fix u and find the minimum value of v,

$$min_v \int_{\Omega} 1/2\theta \, |u-v|^2 + \lambda \, |\rho \, (v)| \tag{2.10}$$

This two-step solution provides an estimate of the optical flow data, which then can be used to analyze the magnitude and direction of relative motion between objects in contiguous frames.

2.3.3 Weakly supervised learning for classification

Weakly supervised learning [1, 18] is a branch of machine learning that deals with this challenge of weakly labeled datasets where the granularity of the labels is low. Weakly supervised algorithms rely on less data point information than supervised algorithms. They use similarity assessments on tuples of data points, such as pairs of similar and dissimilar points, as input instead of labeled points.

Why use weakly supervised learning?

As shown in Figure 2.1 every video in the dataset is divided into non-overlapping snippets. The number of frames in each snippet depends on the architecture of the feature extraction network. Hence, the snippet or snippets containing the anomalous activity would differ based on our network. Furthermore, the most popular datasets do not provide segment-level labels.[19, 20, 21, 22]. It would also be very tedious to label every snippet of all the videos in the dataset. As a result, we rely on weakly supervised learning techniques to train our deep neural network model.Weakly supervised learning techniques also reduce the need for manual annotations for each datapoint. For these reasons we will be using a weakly supervised learning technique known as **Multiple Instance Learning (MIL)**.

Multiple Instance Learning (MIL)

It is a weakly supervised learning technique in which the training data points are grouped in distinct sets called bags [23]. These bags are then provided with labels rather than individual instances. MIL is a very efficient technique for using large datasets without having labels for each data point, reducing the need to label each instance. There are two types of bags in a typical binary classification problem: positive and negative. MIL technique assumes that there are only negative instances in a negative bag, whereas there is at least one positive instance in a positive bag [24]. Consider Figure 2.4 where three people have three keychains, and each keychain has multiple keys. The people holding the keychains only know whether their bunch has the correct key, but they don't know which key opens the door. MIL model works on this concept and figures out that the green key opens the door. The training data in a MIL model is in represented as a set of bags $X_{train\_set} = \{X_1, X_2, X_3, X_4, ...\}$ with labels $Y_{labels} = \{Y_1, Y_2, Y_3, Y_4, ...\}$. The bags

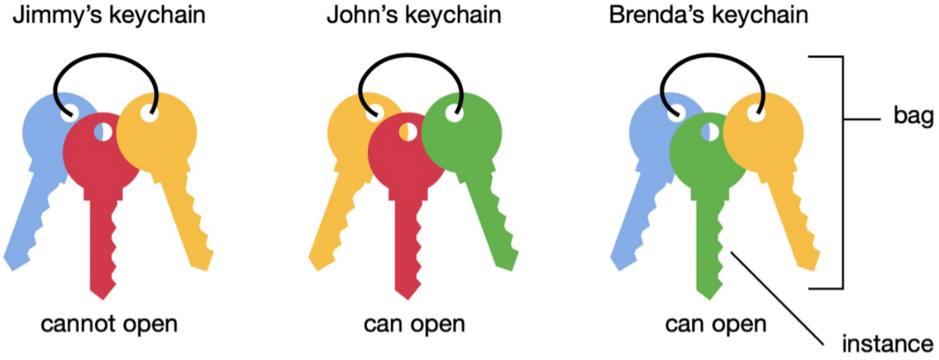


Figure 2.4: An example of the idea behind Multiple Instance learning technique.

contain the actual train samples which are used to train the classifier, each bag is can be represented as, $X_1 = \{x_1, x_2, x_3, x_4, ...\}$. The MIL algorithm tries to find which of the samples from the set X_1 can be mapped to the label Y_1 without the need of instance level labelling.

In MIL, the model learns from both the positive and the negative instances, widening the scope of information that can be processed. Sultani et al. [9]; have trained a deep neural network model based on the MIL technique using the spatio-temporal features. They have used both normal and abnormal videos to train their model. The success of this model led to more techniques developed on the MIL technique. Tian et al. [25] proposed a novel approach based on a variation of the MIL model that they termed Robust Temporal Feature Magnitude learning (RFTM). They have created a Multiscale Temporal Network(MTN) for improving the context in the temporal space. Feng et al. [26] have provided a model with automated annotation generation to tackle the problem of manual annotations. They combine a pseudo-label generator with Self Guided Attention (SGA) encoder on the MIL model to create an automated video anomaly detection system. After observing the recent trends in this domain, we decided to use the MIL technique to train our deep learning network.

CHAPTER 3
Proposed work

As shown in Figure 2.1, the two principle modules of a video anomaly detection system are the feature extractor and the classifier. In our model, we use the two-stream I3D network for feature extraction, then use these feature vectors to train the MIL network using the ranking loss criteria. We propose a modification to the standard ranking loss function specific to the problem of anomaly detection of videos. The MIL network scores every video on a scale of 0 to 1. The higher the score greater the chances of finding the anomaly in that video. We now discuss the various modules of the proposed architecture.

3.1 Feature extraction using a two-stream I3D network

The I3D model we use is a composite of RGB and Optical flow data of contiguous video frames. The RGB features are extracted from the same I3D network described in Section 2.1.1. For extracting the Optical flow features, we first extract the Optical flow data from consecutive video frames by the TV-L1 [17] algorithm. The Optical flow data is then passed through another I3D network, and the features from the RGB and Optical flow are concatenated to create a composite feature vector. The 1024 features from both are concatenated to create a final feature vector of 2048 features. We use this composite feature calculated for each snippet to train the MIL network. Figure 3.1 shows the process feature extraction from the two-stream network. It is important to note that we do not perform a softmax classification to detect the action in the video. The original I3D network was created to recognize human actions and classify them. Instead, in our case, we directly use the output fully connected layer as input to our classification network.

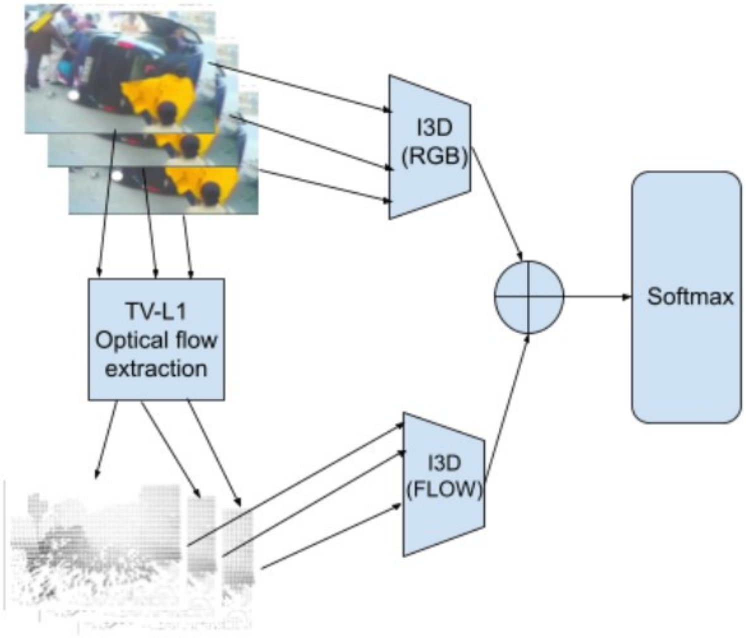


Figure 3.1: The working of the two-stream I3D network

3.2 The MIL network for video classification

The data needs to be split into positive and negative bags for training the MIL model. In the case of video anomaly detection, a positive bag is the set of snippets in which at least one snippet is from an anomalous video, and a negative bag has only snippets from normal videos. As explained in section 3.1 we have extracted the two-stream I3D features, which are then used to train the fully connected MIL neural network. The anomaly video (positive bag) and the normal video (negative bag) are divided into non-overlapping contiguous video segments (instance of the bag). A video is thus a collection of N such snippets V_i where i is $\{1, 2, 3, ...N\}$. Any video is thus represented as $v = \{V_i\}_1^N$ is labelled as a positive bag B_a if at least one of the N-snippets is anomalous and is negative bag B_n otherwise. y is the label for each bag and $y = 1$ for a B_a or $y = 0$ for B_n. Both these bags are passed through a function $f(x)$ which then provides a score for each snippet, a max-pooling is performed to determine the highest score, and then that bag is scored as per the max value. This score per bag is used as a parameter for ranking loss calculations. The objective behind using the ranking loss is to ensure that the abnormal video snippet gets a higher score than the normal.

3.2.1 Ranking loss

Ranking loss is based on the process of metric learning. It aims at predicting the relative distances between different inputs. Ranking loss functions are pretty flexible in training data: We only require a similarity score between data points to use them. The score might be either binary (similar or dissimilar). Consider a face authentication problem in which we know which photographs belong to the same person (similar) and which do not (dissimilar). We train our classifier to recognize the similarities between the patterns in an anomaly video and a normal video by using a ranking Loss function. To apply a ranking Loss function, we must first extract features from two (or three) input data points and create an embedded representation for each. Then we define a metric (anomaly score in our case) for each of the inputs and then try to maximize the distance between them. By back-propagation, we adjust the weights in the hidden layers to achieve the desired output. The standard ranking loss for a pair of positive and negative samples is given by,

$$L = max(0, m - d(r_p, r_n)) \tag{3.1}$$

Where $d(r_p, r_n)$ is the euclidean distance between the positive r_p and negative r_n sample's metric values. m is the max value of the metric. For our case, we have defined the positive and negative samples as bags, and the max value of our metric, the anomaly score, is 1. Thus in our case, the loss is given by,

$$L = max(0, 1 - f(B_a), +f(B_n)) \tag{3.2}$$

Where $d(r_p, r_n)$ is the euclidean distance between the positive r_p and negative r_n sample's metric values. m is the max value of the metric. For our case, we have defined the positive and negative samples as bags, and the max value of our metric, the anomaly score, is 1. Thus in our case, the loss is given by,

$$L = max(0, 1 - f(B_a), +f(B_n)) \tag{3.3}$$

where $f()$ scores the bags using the two-stream composite I3D features. The loss function reaches its minmum value when the value of $f(B_a)$ approches 1 and $f(B_n)$ reaches 0.

3.2.2 The architecture MIL network

We have created a MIL deep neural network using the two-stream I3D features as training data and ranking loss function. The input to this network is pair of videos (normal and anomaly) represented as collection composite I3D feature vectors. Each vector represents a segment of the video of fixed length. The network then scores each of these snippets and the bags (positive and negative videos) based on the max value of the snippets. Then as discussed in section 3.2.1, the ranking loss function is used for back-propagation to achieve the desired goal.

Sultani et al.[19] have noted some shortcomings of the loss calculated by Equation 3.3. They explain that an anomaly occurs for a short duration of time; hence, the occurrences of anomalous instances in the negative bag would be sparse. They have also proposed that there is an abrupt change in the anomaly score in consecutive segments when an anomaly occurs. As a result, they have added sparsity and smoothness constraints to their loss function to adjust for irregularities. We also use these constraints in our loss function, which then is given by,

$$sparsity = \lambda_1 \sum_{i}^{n} f(V_a^i) \tag{3.4}$$

$$smoothness = \lambda_2 \sum_{i}^{n-1} \left(f(V_a^i) - f(V_a^{i+1})\right)^2 \tag{3.5}$$

Where λ_1 and λ_2 are sparsity smoothness and constants respectively and V_i is the snippet with max anomaly score selected from the anomalous video, it is also important to note that the constraints are added only for the anomalous instances of the positive bags. On adding these constraints to the loss function of Equation 3.3 we get the final MIL ranking loss function as ,

$$L = max(0, 1 - f(B_a), +f(B_n)) + sparsity + smoothness \tag{3.6}$$

Figure 3.2 illustrates the entire flow of training the above mentioned MIL model. Starting from the I3D composite feature extraction network to the scoring function and ending at the ranking loss calculation at each training epoch. Once trained the threshold value is calculated and used as a classification parameter.

3.2.3 The modified loss function

The model discussed above is trained on the general ranking loss criteria. This model can produce two types of false alarms viz,

1. **a normal video is categorized as anomaly**
2. **an anomaly video is labeled as normal**

An effective anomaly detection system must counter both false alarms. Furthermore, we also need to consider the presence of multiple anomalous snippets, which Equation 3.6 does not factor in. To tackle the first challenge, we ensure the tightest upper bound on the anomaly score of a normal video, thus reducing the chances of predicting a normal video as anomalous. Hence we perform max-pooling at the final stage when all the segments from both the bags are scored. In order to mitigate the second type of false alarms and consider the presence of multiple anomalous snippets, we propose a modified loss function.

In the modified ranking loss function, we take the anomaly scores of all the snippets and sort them in descending order. Then we take the top k scores and compare these values with the highest value of the anomaly score of a normal video. The loss value is calculated for all the k snippets, and the net loss is a weighted average of all the loss values. The aim here is to maximize the difference between the normal and abnormal instances and cover the maximum possible anomalous snippets extending our temporal space. Consider the an anomaly video as a collection of N snippets, and $f(x)$ the scoring function. Thus a video v_a can be thus represented as a collection of anomaly scores as follows,

$$v_a = [f(V_a^1), f(V_a^2), f(V_a^3).....f(V_a^n)] \tag{3.7}$$

Now we sort these scores in descending order and select top k scores where $k < n$,

$$v_{a_{sorted}} = [f(V_{a_{sorted}}^1), f(V_{a_{sorted}}^2), f(V_{a_{sorted}}^3).....f(V_{a_{sorted}}^k)] \tag{3.8}$$

henceforth the anomaly score for i^{th} snippet of an anomaly video $f(V_{a_{sorted}}^i)$ is represented as r_a^i and the similar for a normal snippet as r_n^{max} , then we calculate the loss value for each of $i \epsilon = \{1, 2, 3, ...k\}$ by using Ranking loss criteria as in Equation 3.3

$$l_i(r_a^i, r_n^{max}) = max(0, 1 - r_a^i + r_n^{max}) \tag{3.9}$$

After this we take a weighted average of all these k losses to calculate the net loss,

$$net\_loss = \sum_{i=0}^{k} (\alpha_i * l_i(r_a^i, r_n^{max})) \tag{3.10}$$

where, $\sum_{i=0}^{k} \alpha_i = 1$. This is the final loss value we evaluate at each iteration. The modified loss function aimed to introduce even more information from the temporal dimension by considering multiple instances. Another aspect was to consider the possibility of having multiple anomaly events from the video and have the ability to mark all of them. The loss function in Equation 3.6 only considers anomalous segments adjacent to the highest one. In contrast, we consider the maximum possible segments in the modified loss function by taking top k snippets. We have taken the weighted average to ensure that the overall loss value never exceeds 1. The weights are assigned equally as $1/k$ then, by experimentation, we find values of α_i for which the model gives best accuracy.

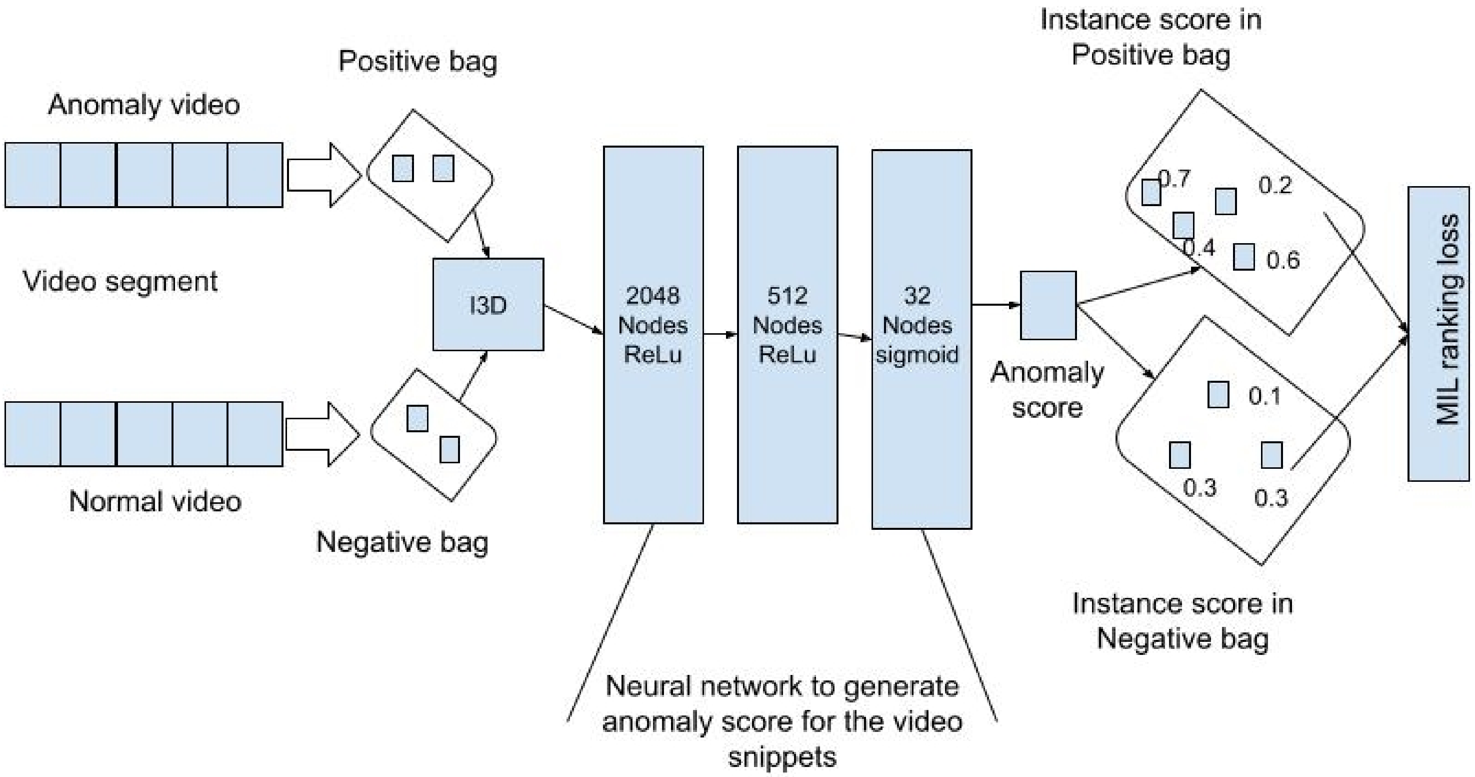


Figure 3.2: The training process of the proposed MIL model with ranking loss function.

CHAPTER 4

Experiment and results

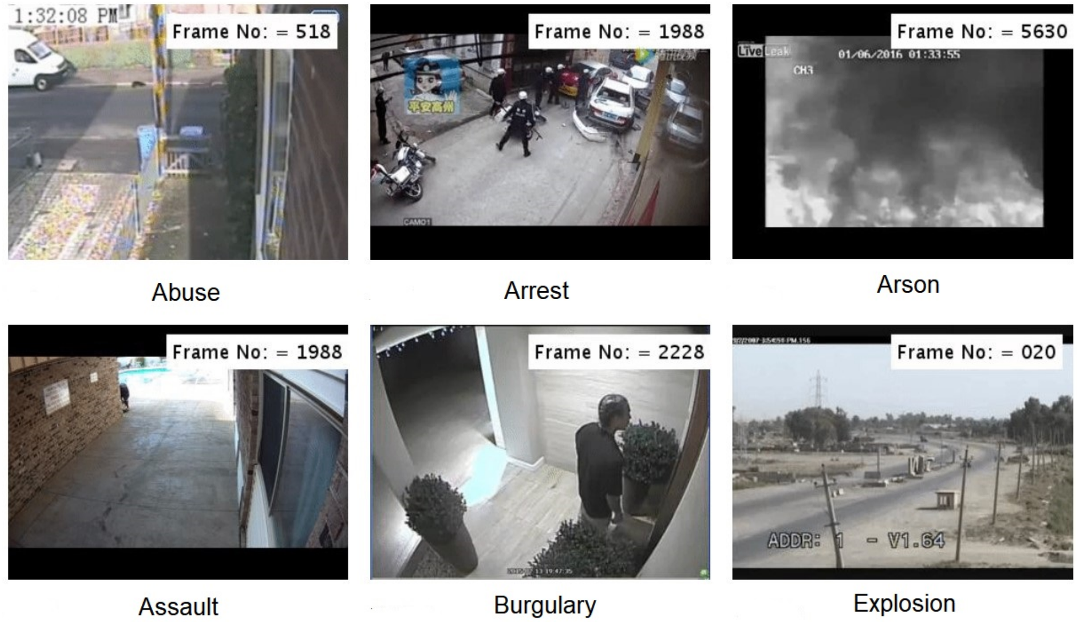
After creating the proposed network, we need to train and test it to check its effectiveness. To check the model's effectiveness, we also need a parameter for benchmarking and compare it with the most recent approaches.

4.1 Dataset

The most widely used datasets for video anomaly detection are UCSD Ped1, Ped2, abnormal crowds, UMN, Avenue [19, 20, 21, 22]. These datasets contain a very small number of video samples in a very limited range of environments. Therefore they are not ideal for creating generalized anomaly detection systems for practical applications. Table 4.1 describes the various types of datasets for video anomaly detection. The datasets are compared for their size, and the types of anomalies covered. We have used the UCF-Crime[19] dataset for training and testing the model. This dataset is a compilation of 128 hours of surveillance camera footage of 13 types of abnormal incidents like Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. It also has normal videos in equal numbers of similar environments. Figure 4.1 has some snapshots of few types of anomalies of the UCF-Crime dataset. The dataset is divided into 1610 training and 290 testing videos. The train set comprises 800 normal and 810 abnormal videos; the test set is split into 150 normal and 140 abnormal instances. Figure 4.2 describes the various types of anomalies and the number of videos of each type. Although the dataset provides a large number of videos for each type of activity, we do not distinguish between the activities. Our objective is to identify the presence of abnormal activity in the video. Hence we ignore the activity specific labels and label each anomalous video as 1 and normal video as 0. While testing the model, we follow the same convention. The aim is to detect the presence of anomalous activity in the abnormal video; such detection is our True Positive instance. For the

Table 4.1: Comparison of various video anomaly detection datasets.

Name	# of video	Average # of frames	Dataset length	Example anomalies
UCSD Pedestrian 1	70	201	5 mins	Bikers, small carts, walking across walkways
UCSD Pedestrian 2	28	163	5 min	Bikers, small carts, walking across walkways
UMN	5	1290	5min	Running
Abnormal Crowd	31	1408	24 min	Panic, fight,congestion, obstacle, neutral
UCF-Crime	1900	7247	128 hours	Abuse, arrest, arson, assault, accident, burglary, fighting, robbery



Abuse

Arrest

Arson

Assault

Burgulary

Explosion

Figure 4.1: A sample snapshot from video snippets of various types of anomalies of the UCF-Crime dataset.

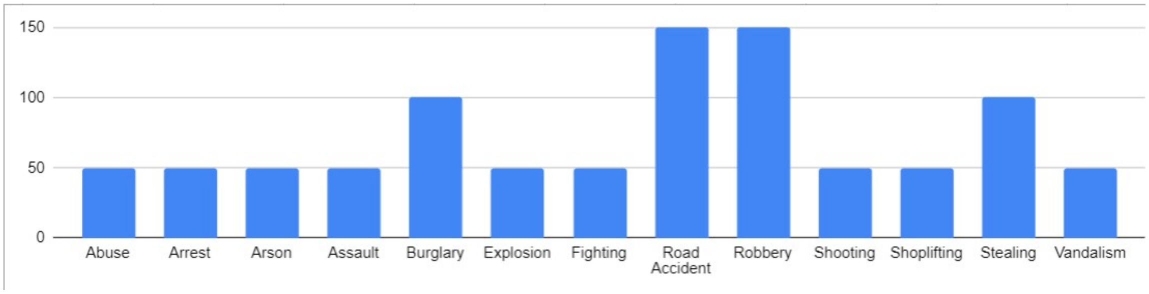


Figure 4.2: The total number of samples of each type of anomaly adding to 950 in UCF-Crime dataset.

anomaly classification problem we classify the anomalies of 13 classes, labelled as their names. For testing the anomaly classification we use the 140 anomaly samples from the test set.

4.2 Implementation details

For the implementation, each video is converted into the 240x320 format at 30fps and then divided into 16 frame clips each. We use the standard weights for the I3D network available as *'mixed.c'*. The features are calculated for each 16-frame clip and then aggregated to create video segments. For our implementation, we divide each video into 32 segments of equal length depending on the size of video. After passing these segments through a fully connected I3D network, as discussed in 3.1, we obtain a 2048D vector of composite features. The feature vector is used as input in the first layer having 2048 nodes; this layer is connected to a 512 node hidden layer with a ReLu activation function and a dropout rate of 0.6. The next layer is of 32 nodes with the same activation function. The final anomaly score is obtained by flattening the output of the 32 nodes via a sigmoid function. For optimization, we have used Adagrad optimizer [28] with a learning rate of 0.001. To calculate the loss value we have set $k = 5$ and the values of $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\} = \{0.4, 0.3, 0.1, 0.1, 0.1\}$. These values for $k = 5$ gave the best accuracy and were found experimentally. The hyperparameters of λ_1 and λ_2 used as sparsity and smoothness constraints are set to 0.00008. We used these parameter values based on the observations provided by Sultani et al. [19]. For preprocessing the videos, we use the OpenCV and PyTorch to create the neural network model. We have trained the setup on the Google Colab platform using Tesla K80 GPUs for 200 epochs.

4.3 Results and analysis

The loss values calculated at each iteration for the original and modified loss $(k = 5)$ functions were compared. It was observed that the modified loss function resolves faster, resulting in less training time by reaching the optimum minimum in lesser epochs. Figure 4.3 is a plot of of the loss values for both the loss functions over 200 epochs. At 50 epochs, the loss value of the modified loss function is about 9.60% lower. We observed that the final loss value of the modified loss function is 6.5% lower than that of the original loss function.

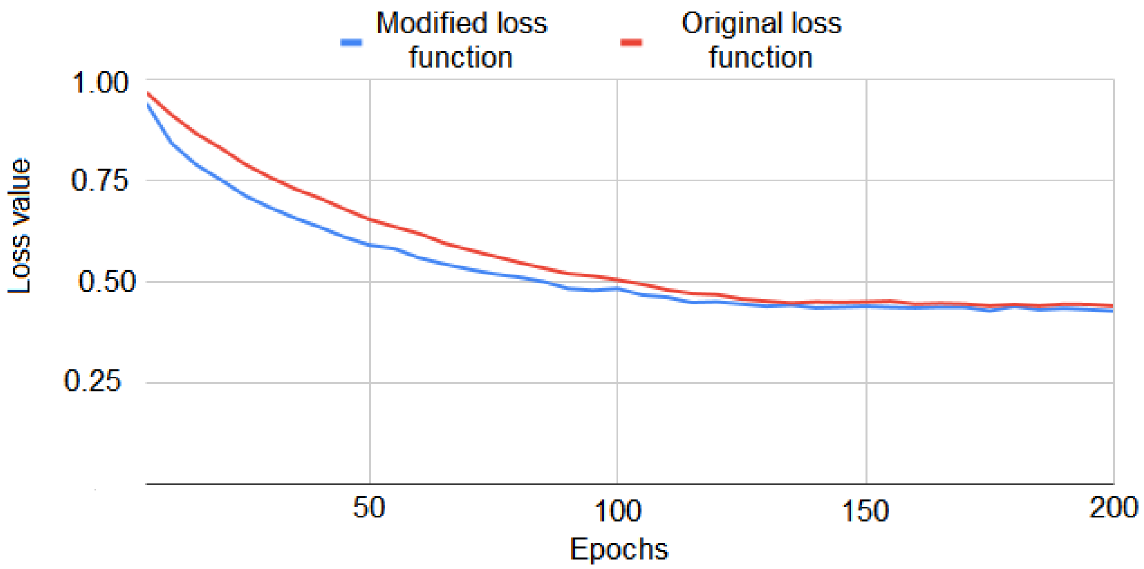


Figure 4.3: The value of loss function after each epoch for both the original and modified ranking loss function ($k = 5$).

4.3.1 The Receiver Operating Characteristic (ROC) curve

To evaluate the model, we use the AUC value of the ROC plot for the test set results. ROC curve is a plot of TPR(True Positive Rate) on the Y-axis and FPR(False Positive Rate) on the X-axis for evaluating the performance of models at various thresholds(FPR values). To calculate TPR and FPR, we calculate the number of True Positives, True Negatives, False Positives, and False Negatives. When a test case has an anomaly, and the model detects it, that is a True Positive (TP). If the test case does not have an anomaly, but the model detects one, it is a False Positive (FP). Similarly, we have a True Negative (TN) for the correct identification of a normal case and a False Negative (FN) for an incorrect identification. Using these terms, TPR and FPR are defined as follows:

$$TPR = TP/(TP + FN) \tag{4.1}$$

$$FPR = FP/(FP + TN) \tag{4.2}$$

After the ROC graph is plotted, we calculate the AUC value, which is a benchmark of the model's effectiveness in classifying the test cases. The value of the area under the ROC curve is always in the range of [0,1]; the higher the value, the better the model at predicting the label. The AUC value shows the measure of separability between the classes. A value close to 1 means the separability is excellent, 0 means the classification is precisely the opposite (all samples with label 'A' is labeled as 'B' and all samples with label 'B' as 'A'). An AUC value of 0.5

indicates that the model has learned nothing and has no criteria for classification.

The ROC plot for Sultani et al. [19], and our proposed model is compared in Figure 4.4. We also compare the AUC values with some recently reported techniques used for video anomaly detection in Table 4.2. We use the results that have been reported by various publications [9, 11, 19, 25, 29] for these methods.

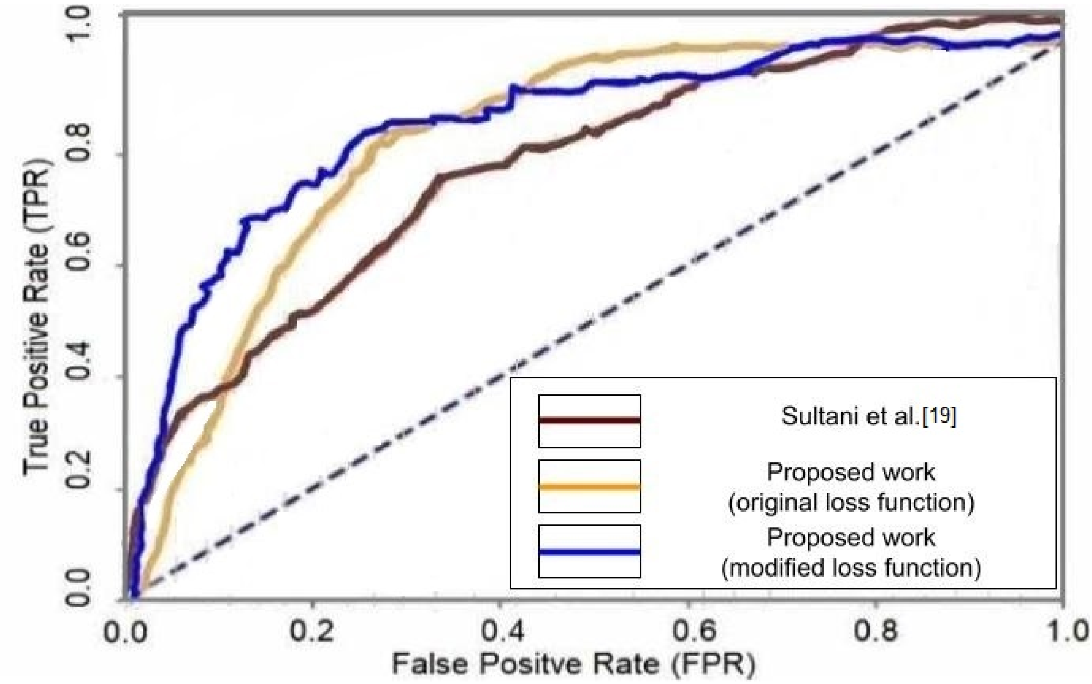


Figure 4.4: ROC plot of Sultani et al. (maroon), our model with original and modified loss function (yellow and blue respectively).

Table 4.2: Comparison of AUC values of the ROC curve for UCF-Crime dataset

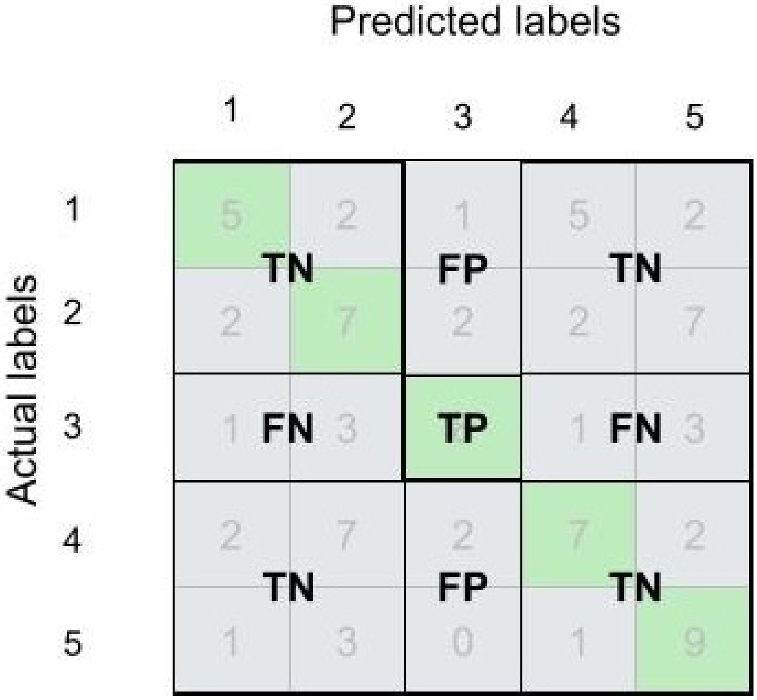
Method	**AUC** (%)
Lu et al. [9]	65.51
BODS [29]	68.26
GODS [29]	70.46
GMM based [11]	75.90
Sultani et.al [19]	77.92
Proposed model	**82.03**
Proposed model (modified loss function)	**84.11**

We have compared our results with popular unsupervised and semi/ weakly supervised techniques used for anomaly detection. Lu et al. [9] had based their method on a dictionary-based technique to understand normal behavior and used the reconstruction approach to detect the presence of anomalies. BODS (Basic

One-class Discriminative Subspaces) and GODS (Generalized One-class Discriminative Subspaces) are unsupervised clustering techniques for anomaly detection given by [29]. Another clustering technique based on GMM (Gaussian Mixture Model) based on Bayesian distribution is also provided by [11]. Our proposed model with the modified loss function provides a better AUC value by a factor of 10.81% from the highest value for the unsupervised learning technique. We also compare our results with the MIL technique introduced by Sultani et al. [19] and have achieved an improvement of about 7.92% in the AUC metric.

4.3.2 The Confusion matrix

For analyzing the accuracy of the anomaly classification problem, we use the confusion matrix. The confusion matrix is a tabular representation of the performance of a classification model. In our case, we have 13 classes of various anomalies as per the UCF-Crime dataset. Figure 4.5 shows a confusion matrix for a multi-class (5-classes) classification problem where the terms True Positive (TP), False Positive (FP), True Negative (TN), and False Negative(FN) are defined. The numbers represent the number of samples predicted for each label against the actual label. The diagonal (in green) is the number of TPs, FPs, FN, and TN for each class.



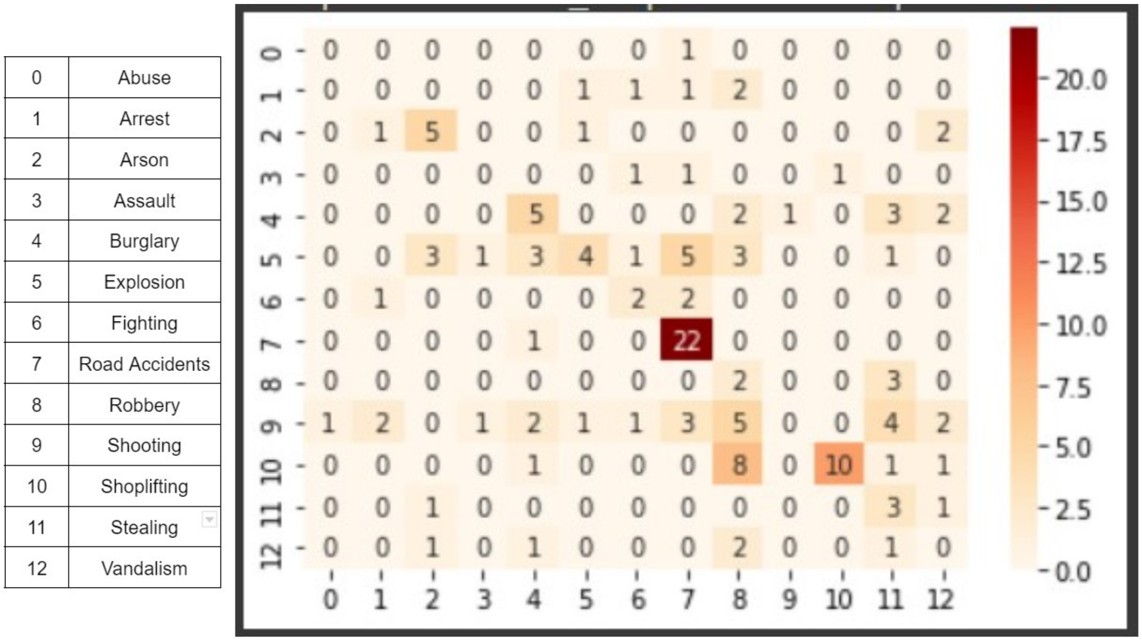
	1	2	3	4	5
1	5	2	1	5	2
2	2	7	2	2	7
3	1	3		1	3
4	2	7	2	7	2
5	1	3	0	1	9

Figure 4.5: Sample confusion matrix the values for class with label 3 are highlighted.

The overall accuracy of the classification model is defined as,

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4.3}$$

The confusion matrix for the video anomaly classifcation is shown in Figure 4.6. The total number of test samples were 139 out of which 53 where correctly classified with an accuracy of 38.12% . Table 4.3 compares the overall accuracy of our model with that of Sultani et al.[19]. The proposed model was able to give an accuracy value which was 34.22% higher than the comapared methods.



0	Abuse
1	Arrest
2	Arson
3	Assault
4	Burglary
5	Explosion
6	Fighting
7	Road Accidents
8	Robbery
9	Shooting
10	Shoplifting
11	Stealing
12	Vandalism

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	0	0	0	0	1	1	1	2	0	0	0	0
2	0	1	5	0	0	1	0	0	0	0	0	0	2
3	0	0	0	0	0	0	1	1	0	0	1	0	0
4	0	0	0	0	5	0	0	0	2	1	0	3	2
5	0	0	3	1	3	4	1	5	3	0	0	1	0
6	0	1	0	0	0	0	2	2	0	0	0	0	0
7	0	0	0	0	1	0	0	22	0	0	0	0	0
8	0	0	0	0	0	0	0	0	2	0	0	3	0
9	1	2	0	1	2	1	1	3	5	0	0	4	2
10	0	0	0	0	1	0	0	0	8	0	10	1	1
11	0	0	1	0	0	0	0	0	0	0	0	3	1
12	0	0	1	0	1	0	0	0	2	0	0	1	0

Figure 4.6: The confusion matrix for anomaly classification model.

Table 4.3: Comparison of the overall classification accuracy of the proposed model with state of the art methods.

Method (Feature extraction)	**Accuracy** (%)
Sultani et al (C3D). [19]	23.0
Sultani et al (TCNN) [19]	28.4
Proposed model (I3D composite)	**38.12**

CHAPTER 5
Conclusion and future work

5.1 Conclusion

In this thesis, we have done an exhaustive study on the problem of video anomaly detection and the various techniques employed to tackle it. We looked at the traditional hand-crafted methods, which focused on trajectory analysis and extracting low-level features to detect abnormal patterns in crowds. It was observed that these methods were compute intensive, and there was a need for automated feature extraction. So high-level feature extraction techniques were developed based on the low-level features. The most recent approach for solving this problem was deep learning. CNN-based feature extraction methods were found to be the best for analyzing video frames. For training the deep learning classifier, we discovered that weakly supervised learning techniques were most practical due to the nature of the data. Manual annotation of every abnormal video segment was no longer required.

Based on the survey done, we have created a deep learning solution based on the MIL ranking loss model and two-stream I3D network. The RGB and optical flow features were concatenated to create a composite feature vector. Due to the range and complexity of real-world scenarios, we use both the abnormal and normal videos in the training set. We tested our implementation on the UCF-Crime dataset and evaluated the AUC of ROC curve values. The proposed method has achieved better results than some of the recent approaches on the AUC metric. We have provided a modified ranking loss function to cover multiple anomalous instances in each video. The modified loss function performs better than the standard loss function reaching the minimum value in fewer epochs. In the anomaly classification problem, we achieved an accuracy that was 34.22 % higher than reported by Sultani et al [19].

5.2 Future work

The two-stream model of generating composite feature vector can be merged into a single step process rather than concatenating the output of two I3D networks. Although we do not need the segment-level labels in our model, we need to create an efficient automated annotation technique. An automated annotation network would make it possible to apply supervised deep learning algorithms, which are known to perform better than the unsupervised or weakly supervised ones. We can improve the accuracy of the anomaly classification problem by including location-specific details in the feature vector. For example, a bank robbery and a house burglary are distinguishable by location. The action recognition model cannot make this distinction. Furthermore, we need to have specific datasets for each type of location for different anomalies.

References

[1] Nayak, R., Pati, C., A comprehensive review on deep learning-based methods for video anomaly detection, Image and Vision Computing, Volume 106, 104078 (2021), doi: 10.1016/j.imavis.2020.104078.

[2] Yang, B., Cao, J., Anomaly detection in moving crowds through spatiotemporal autoencoding and additional attention, Advances in Multimedia 1–8 (2018). doi: 10.1155/2018/2087574 .

[3] Zhao, Y., Deng, B., Spatio-temporal autoencoder for video anomaly detection, Proceedings of the 25th ACM International Conference on Multimedia, 1933–1941 (2017), doi: 10.1145/3123266.3123451.

[4] Fernyhough, J., Cohn, A., Generation of semantic regions from image sequences. In: Buxton, B., Cipolla, R. (eds) Computer Vision — ECCV '96. ECCV 1996. Lecture Notes in Computer Science, vol 1065. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-61123-1_162.

[5] Tung,F., Zelek, J., Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. Image and Vision Computing, 29(4):230-240, 2011, https://doi.org/10.1016/j.imavis.2010.11.003.

[6] Calderara, S., Heinemann, U., Detecting anomalies in people's trajectories using spectral graph analysis. Computer Vision and Image Understanding, 115(8):1099-1111, 2011, https://doi.org/10.1016/j.cviu.2011.03.003.

[7] Adam,A.,Rivlin,E.,Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555-560, March 2008, doi: 10.1109/TPAMI.2007.70825.

[8] Wang, S., Zhu, E., Video anomaly detection and localization by local motion based joint video representation and OCELM, Neurocomputing, Volume 277, 2018, Pages 161-175, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.201.08.156.

[9] Lu, C., Shi, J., Abnormal Event Detection at 150 FPS in MATLAB, IEEE International Conference on Computer Vision, pp. 2720-2727 (2013). doi: 10.1109/ICCV.2013.338.

[10] Hasan, M., Choi, J., Learning Temporal Regularity in Video Sequences, Computer Vision and Pattern Recognitions(CVPR), 733-742 (2016), doi: 10.1109/CVPR.2016.86.

[11] Degardin, B., Weakly and Partially Supervised Learning Frameworks for Anomaly Detection (2020), doi: 10.13140/RG.2.2.30613.65769.

[12] Kiran, B., Thomas, D., An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos, Journal of Imaging 4 (2), 36 (2018), doi: 10.3390/jimaging4020036.

[13] Liu, Y., Li, Z., Generative adversarial active learning for unsupervised outlier detection, IEEE Transactions on Knowledge and Data Engineering 32 (8), 1517–1528 (2020) doi: 10.1109/TKDE.2019.2905606.

[14] Tran, D., Bourdev, L., Learning Spatiotemporal Features with 3D Convolutional Networks, International Conference on Computer Vision (ICCV), 4489-4497 (2015), doi:10.1109/ICCV.2015.510.

[15] Carreira, J., Zisserman, A., Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset,CVPR, 4724-4733 (2017) doi: 10.1109/.2017.502.

[16] Kay, W., Carreira, J., The Kinetics Human Action Video Dataset CVPR, (2017), doi: 10.48550/ARXIV.1705.06950.

[17] Sanchez, J., Meinhardt-Llopis, E., TV-L1 Optical Flow Estimation, Image Processing On Line, 3 (2013), pp. 137–150. https://doi.org/10.5201/ipol.2013.26

[18] Popoola, O.,Wang, K., Video-based abnormal human behavior recognition-a review, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42 (6) 865–878 (2012), doi: 10.1109/TSMCC.2011.2178594.

[19] Sultani, W., Chen, C., Real-World Anomaly Detection in Surveillance Videos, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 6479-6488 (2018). doi: 10.1109/CVPR.2018.00678

[20] Li, W., Mahadevan, V., Anomaly detection and localization in crowded scenes, IEEE Trans Pattern Anal Mach Intell, Jan;36(1):18-32 (2014). doi: 10.1109/TPAMI.2013.111.

[21] Mehran, R., Oyama A., Abnormal crowd behavior detection using social force model, IEEE Conference on Computer Vision and Pattern Recognition, pp. 935-942 (2009), doi: 10.1109/CVPR.2009.5206641.

[22] Rabiee, H., Haddadnia, J., Novel dataset for fine-grained abnormal behavior understanding in crowd, 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 95-101 (2016), doi: 10.1109/AVSS.2016.7738074.

[23] Maximilian, I., Jakub, M., Attention-based Deep Multiple Instance Learning, International Conference on Machine Learning (ICML) , PMLR 80:2127-2136, (2018).

[24] Bing, L., Wang, W., Sparse Representation Based Multi-Instance Learning for Breast Ultrasound Image Classification, Computational and Mathematical Methods in Medicine,1-10 (2017), doi: 10.1155/2017/7894705.

[25] Tian, Y., Pang, G., Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 4975-4986 (2021), doi: 10.48550/arXiv.2101.10030.

[26] Feng, J., Hong, F., MIST: Multiple Instance Self-Training Framework for Video Anomaly Detection, Conference on Computer Vision and Pattern Recognition (CVPR), (2021), doi: 10.1109/CVPR46437.2021.01379.

[27] Nguyen, T., Meunier, J., Anomaly Detection in Video Sequence With Appearance-Motion Correspondence, IEEE/CVF International Conference on Computer Vision (ICCV), 1273-1283 (2019). doi: 10.1109/ICCV.2019.00136.

[28] Duchi, J., Hazan, E., Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research. 12. 2121-2159 (2011).

[29] Wang, J., Cherian, A., Gods: Generalized one-class discriminative subspaces for anomaly detection, IEEE International Conference on Computer Vision, pages 8201–8211, (2019), doi: 10.1109/ICCV.2019.00829.