ANOMALY DETECTION IN LIVE SURVEILLANCE VIDEOS

by

K. JAYA SHANKAR REDDY	***411529***
G. NAVEEN REDDY	***411525***
G. DEVA RAJ	***411523***

Under the esteemed guidance of

Dr. K Hima Bindu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGYANDHRA PRADESH

TADEPALLIGUDEM-534102, INDIA

May - 2019

ANOMALY DETECTION IN LIVE SURVEILLANCE VIDEOS

Thesis submitted to

National Institute of Technology Andhra Pradesh

for the award of the degree

of

Bachelor of Technology

by

K. JAYA SHANKAR REDDY	***411529***
G. NAVEEN REDDY	***411525***
G. DEVA RAJ	***411523***

Under the esteemed guidance of

Dr. K Hima Bindu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGYANDHRA PRADESH

TADEPALLIGUDEM-534102, INDIA

May - 2019

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

K. Jaya Shankar Reddy	G. Naveen Reddy	G. Deva Raj
Roll No: 411529	Roll No: 411525	Roll No: 411523
Date:	Date:	Date:

CERTIFICATE

It is certified that the work contained in the thesis titled "**Anomaly Detection in Live Surveillance Videos**," by "K. Jaya Shankar Reddy, bearing Roll No: 411529" has been carried out under my supervision and that this work has not been submitted elsewhere for a degree

Dr. K. Hima Bindu
Dept. of Computer Science and Engineering
N.I.T. Andhra Pradesh
May, 2019

ACKNOWLEDGMENTS

First and for most, I would like to thank my guide Dr. K Hima Bindu for guiding me thoughtfully and efficiently through this project, giving me an opportunity to work at my own pace along my own terms, while providing me with very useful directions whenever necessary.

Also, I would like to thank Dr. Nagesh Bhattu for showing interest in my project and motivating me by providing useful inputs towards the successful completion of the project. I am grateful to Dr. Gautham Reddy for providing the computer lab facility whenever necessary for the resource intensive computations involved in my project.

I would also like to thank my friends Naveen Reddy and Deva Raj for helping me and providing encouragement throughout the length of this project

I offer my sincere thanks to all other persons who knowingly or unknowingly helped me complete this project.

LIST OF FIGURES

LIST OF TABLES

NOTATIONS

Notation	Meaning
MIL	Multiple Instance Learning
Segment or sequence of a video	A set of continuous frames
CCTV	Closed Circuit Television
C3D	Convolution 3 Dimensional Network
Trimmed Video or Temporal Annotation of video	Starting and ending frame numbers of an anomaly
Positive class and Negative class	Anomalous and Normal class
Bags	Videos
SSMIL	Similarity based Semi-supervised Multiple Instance Learning
ConvNet	Convolution Network
w	Model to be learned
f	Anomaly score that is given to a segment
FC layer	Fully Connected Layer
l	Value of loss function
ROC curve	Receiver Operating Characteristic
AUC	Area Under the Curve of ROC

ABSTRACT

There are millions of CCTVs deployed in and around the world. They capture the scene or activity happening in front of them and there are now millions of footages. People committing crime are aware of the CCTVs but yet they commit because there is no one to take immediate action on them. If it is possible to analyze the video in real time, then a notification can be sent to the authorities immediately regarding the crime. But the human effort is required to analyze the live footage or to find a particular activity/person in the pre-recorded footages. It is a very tiresome process and humans are prone to make mistakes. Hence, it is necessary to develop a system that can automatically detect crime or anomaly in real time and can notify the authorities. Many researchers have proposed methods based on machine learning, object detection, e.t.c. But the problem in those methods is they require huge training data and to prepare such data, again it requires human effort to label the videos either normal or anomalous. Hence, a semi-supervised deep neural network model is proposed where the entire training data need not be labeled and in the part of the data that is labeled, there are no strict restrictions, i.e. video level labels are to be given rather than the individual segment level labels. MIL (Multiple Instance Learning) based ranking loss function is used for the labeled training data and similarity based loss function for the unlabeled training data. Both the losses add up to the error of the neural network. Hence, the proposed model is called the Similarity based Semi-supervised Multiple Instance learning (SSMIL). C3D feature extraction model is used to extract the fixed length features of a video which takes care of both spatial and temporal correlation between the frames of the video. AUC is used as the metric to measure the performance of the model. The proposed SSMIL model performed better than the existing state of the art methods when evaluated using the same dataset.

TABLE OF CONTENTS

Contents

CHAPTER 1

INTRODUCTION

It is estimated that there are around 250 million cameras in use today, generating over 1 billion gigabytes of data every day. Different kinds of activities are being captured by them. They are able to capture a variety of realistic anomalies specifically crime sequences. These CCTV footages are continuously monitored by the staff and report to the higher authorities if there is an occurrence of a crime or anomaly in the live footage so that an action can be taken. The monitoring task is very tedious for the staff and they can't monitor continuously as they take breaks in between. Also, they might want to analyze the previously recorded footages to find a particular person involved in the anomalous activity. They have a huge amount of footages and it is almost impossible to check for anomalies in them.

Hence, it is necessary to automatically detect the anomalous activities in the CCTV footages. Anomaly detection is the identification of rare items, events or observations which raise suspicions differing significantly from the major of the data. Anomalies are also referred to as outliers, novelties, noise, deviations and exceptions. An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism. There are three broad categories of Anomaly detection: Unsupervised anomaly detection, supervised anomaly detection and semi supervised anomaly detection.

Videos are nothing but a sequence of frames. A frame is nothing but a single image. Videos are characterized by the speed with which these frames come one after the other, also called the frames per second.

Anomaly detection in videos is a very critical task because there is a need to specify the starting point and ending point of the frame, i.e. when exactly an anomaly starts and when the anomaly ends. This cannot be done using starting time and ending time because there might be 30 frames in a second on average. Hence, an anomaly is generally specified with a starting frame number and an ending frame number. So, to call a video anomaly, right from the starting frame to the ending frame must be containing anomaly activity. Such videos are called trimmed videos where the entire video contains an anomaly.

There are three different challenges to be addressed for Anomaly detection in Videos. One of them is the unavailability of huge labeled and trimmed training video data. The second is labeling a video sequence always an anomaly. But, in reality, a video sequence treated as an anomaly in one scenario cannot be treated as an anomaly in another scenario. For example, crowd gathering at an exhibition is not an anomaly but crowd gathering on a National Highway shows some evidence of anomaly. The third is the class Imbalance. There are very less number of anomalous videos but a large number of normal videos.

To address the first challenge, the model needs to perform better even with limited available untrimmed training data, i.e. the video labels are known and the exact starting and ending frame of an anomaly in that video is unknown. For the second problem, different models are to be built for different scenarios. For the third problem to be solved, semi supervised learning can be used with limited labeled data and the remaining unlabeled data.

Hence, a model is needed to be trained for every scenario with limited available untrimmed videos specific to that scenario. This can be achieved through semi supervised multiple instance learning. The training data can be developed for any kind of scenario with minimum effort. All that is required is labels at the video level and there is no need to label each and every sequence/segment (starting frame and ending frame) of the video, i.e. training data consists of labels at the video level but not at the segment level. A segment is nothing but a part of an entire video. The aim of the semi supervised multiple instance learning is to predict the labels at the segment label using the labels at the video level. The model proposed performs better than the existing supervised methods.

CHAPTER 2

LITERATURE REVIEW

Anomaly detection is one of the most challenging and long standing problems in computer vision [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. Many anomalous detection techniques have been proposed to solve problems in different domains such as intrusion detection, fraud detection, and database optimization. Chandola et al. [13] argue that anomaly detection is important due to the fact that the detected anomalies often correspond to actionable information within the application domain.

Although anomaly detection has been studied for a long time, many methods have been developed for specific domains or problems. Surveillance video is tremendously tedious to observe when events that need follow-up have a very low probability. For crowded scenes, this complexity is compounded by the difficulty of regular crowd behaviors. The mission of automatically detecting frames with anomalous or interesting events from extended duration video sequences has disturbed the research community in the last decade. Anomaly detection is very important in crowded scenes. For example, in security applications, where it is hard even for trained personnel to consistently monitor scenes in dense crowds or videos of long duration [14].

The ability to identify anomalies in real-time is very precious, so that suitable actions can be taken as soon as it is detected to avoid or reduce negative consequences. Most of the research efforts are done to replace the necessities of manually identifying anomalous situations, to make an automated video surveillance system [1]. In [14], Yingying Zhu et.al introduced a Context-Aware Activity Recognition and Anomaly Detection in Video. An introduced system aims to dynamically capture the frequent motion and context patterns for each activity class, as well as all pair of classes, from sets of predefined patterns during the learning scheme. The system jointly models both motion and context information which is used for activity recognition and anomaly detection. The explicitly models the spatial and temporal relationships of activities and captures useful spatial and temporal patterns for each pair of interesting activity classes during the learning process. It integrates motion features and various context features into a unified model. With the learned pattern parameters, normality factors are introduced to measure the normalcy of activities based on their motion and context features. Activities with one or more normality factors lower than the predefined thresholds (which can be learned a priori) are considered as anomalies. However, it does not robustly distinguish different kinds of anomalies.

In [15], Weixin Li et.al introduced an Anomaly Detection and Localization scheme in Crowded Scenes. An implemented system takes into the accounts of both appearance and dynamics properties. This information is used to implement 1) a center-surround discriminant saliency detector which is used to generate the spatial saliency scores, and 2) a scheme of normal behavior which used to learn from training data and generates temporal saliency scores. Spatial and temporal anomaly maps are represented at multiple spatial scales, by taking scores of these operators at increasingly larger regions of support. The multiscale scores perform potentials of a conditional random field that ensure the global consistency of the anomaly decision.

In [16], Dan Xu et, al introduced a Video anomaly detection based on a hierarchical activity discovery within spatiotemporal contexts. An introduced mechanism detects the anomalies based on a hierarchical activity-pattern detection framework which systematically considering both global and local spatial and temporal contexts. Under the global context, we discover atomic activity patterns from low-level optical flow features, and the distributions of the atomic activity patterns are modeled for higher-level activity representation. Then, salient activity patterns are discovered under the local context. The discovery is a coarse-to-fine learning process with unsupervised technique for automatically making normal activity patterns at various levels. A unified anomaly energy function is introduced based on these discovered activity patterns to recognize the abnormal level of a video.

In [17], Xuan Mo et.al introduced Adaptive Sparse Representations for Video Anomaly Detection. An implemented system is used for detecting the anomalies involving multiple objects. First, a new joint sparsity technique for anomaly detection that facilitates the detection of joint anomalies involving multiple objects is developed. This extension is highly nontrivial since it leads to a novel concurrent sparsity trouble that it is solved by using a greedy pursuit technique. Second, is the kernelization. The kernelization of the joint sparsity model improves anomaly detection in which linear sparse reconstruction models do not hold directly. The linear sparsity scheme is used to enable superior class separability and hence anomaly detection action. However, it does not update expert training periodically.

Datta et al. proposed to detect human violence by exploiting motion and limbs orientation of people. Kooij et al. [18] employed video and audio data to detect aggressive actions in surveillance videos. Gao et al. proposed violent flow descriptors to detect violence in crowd videos. More recently, Mohammadi et al. [19] proposed a new behavior heuristic based approach to classify violent and non-violent videos.

Researchers in [11,12] used sparse representation to learn the dictionary of normal behaviors. During testing, the patterns which have large reconstruction errors are considered as anomalous behaviors. Due to successful demonstration of deep learning for image classification, several approaches have been proposed for video action classification [20,21]. However, obtaining annotations for training is difficult and laborious, specifically for videos. Recently, [13, 2] used deep learning based auto encoders to learn the model of normal behaviors and employed reconstruction loss to detect anomalies.

Learning to rank is an active research area in machine learning. These approaches mainly focused on improving relative scores of the items instead of individual scores. Bergeron et al. [22] proposed an algorithm for solving multiple instance ranking problems using successive linear programming and demonstrated its application in hydrogen abstraction problem in computational chemistry. Recently, deep ranking networks have been used in several computer vision applications and have shown state-of-the-art performances. They have been used for feature learning [23], highlight detection [1], Graphics Interchange Format (GIF) generation [24], face detection and verification [25], person re-identification [26], place recognition [27], metric learning and image retrieval [28]. All deep ranking methods require a vast amount of annotations of positive and negative samples.

In [29], Label propagation is a well-explored family of methods for training a semi-supervised classifier where input data points (both labeled and unlabeled) are connected in the form of a weighted graph. whenever input dataset exhibits following characteristics - (i) one of the class labels is a rare label or equivalently, class imbalance (CI) is very high, and (ii) degree of supervision (DoS) is very low – defined as fraction of labeled points. In this formulation, objective function is a difference of convex quadratic functions and the constraints are box constraints and this problem is solved using Concave-Convex Procedure (CCCP).

Another method is the deep neural network proposed in [30]. The loss function is SVM - based MIL ranking function where higher scores are given to anomalous video sequences and lesser scores to the normal video sequences. It is a completely supervised model and takes no care of Class Imbalance. A model for anomalous activity recognition is proposed using TCNN feature extraction model [31].

In [32], Convolution LSTM (Long Short Term Memory) based model is trained in order to reconstruct the sequence of input frames and predict the sequence of future frames, i.e. the model has an Encoder, Past Decoder and Future Decoder. The training is done with many normal videos making the model unable to predict or reconstruct anomalous frame sequences. So, the reconstruction error expressed as the MSE (Mean Square Error) when crosses a threshold value, the given input's future sequence of the frames is labeled as anomalous. The problem in this method is it uses only normal videos and in reality, there are huge numbers of normal videos which are very different from each other. So, there is a chance that a new unseen normal sequence of frames gets labeled as anomalous by the model.

In [33], an automatic anomaly detection technique for camera anomaly by image analysis, in order to confirm good image quality and correct field of view of surveillance videos. The technique first extracts reduced-reference features from multiple regions in the surveillance image, and then detects anomaly events by analyzing variation of features when image quality decreases and field of view changes. Event detection is achieved by statistically calculating accumulated variations along the temporal domain. False alarms occurred due to noise are further reduced by an online Kalman filter that can recursively smooth the features.

In [34], the authors proposed a system which will be designed to aid crime detection and prevention by processing video frames in real time. Video retrieval is the process of retrieving important information from videos in order to classify them and use them for various purposes. It works on a broader scale by processing multiple images (frames) at a time. The classification is done according to features which are an integral part of video frames and used for motion detection from a static camera. The paper concentrates on extracting information from video frames and using it for crime prevention purposes. The system will take real time video and find out anomalies by detecting motion. The system will raise an alarm to notify authorities of unwanted presence.

CHAPTER 3

PROPOSED APPROACH

3.1 Problem Definition

A model is to be built for a particular scenario, be it either for anomaly detection in traffic areas or exhibition areas or shopping malls with limited amount of untrimmed training data available, i.e. the training video labels are at the video level but not at the segment level. The model should be capable of providing anomaly scores to the unseen video segments/sequences belonging to a particular scenario for which it was trained for. More is the anomaly score, more is the probability of that segment being an anomalous one and lower is the anomaly score, less is the probability of that segment being an anomalous one.

Untrimmed training data of the scenario can be made easily available by not restricting the starting frame and ending frame numbers of the anomaly occurrence. Hence, an Anomalous video can contain any number anomalies in the entire video without the knowledge of the starting frame and ending frame numbers of those anomalies and a Normal video should not be containing any anomaly.

In this way, a part of training videos are labeled and the remaining part of the training videos can be left unlabeled. Hence, the training data videos can be made available with minimum amount of effort and supervision. The model should be capable of leveraging both the labeled and unlabeled training video and predict the scores for new unseen video sequences. The problem definition can be summarized as follows:

Input: Less number of videos labeled as either normal or anomaly. And the labels are video level and not at segment level.

Processing: Extracting the features from videos and training the model.

Output: Scores for every individual segments of the video. Higher score should imply anomalous segment and lower score should imply normal segment.

3.2 Feature Extraction:

Since the aim is to detect anomaly. Anomalous video sequence is not just about a single frame, it is the sequence of frames which makes it anomaly. So, the features extracted should represent a sequence of frames of a video. The sequence of frames is nothing but the segment. The features of the segment should reflect its spatial and temporal features.

Spatial features tell about the behavior of a particular frame where as temporal features tell about the behaviour of sequence of frames.

3.2.1 C3D Model

A model that extracts such features is the C3D, i.e. Convolution 3D Network. C3D is the modified version of the BVLC CAFFE. Berkley Vision and Computer Centre's CAFFE (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework that supports architectures geared towards image classification and image segmentation. It supports GPU and CPU based acceleration computational kernel libraries like NVIDIA CUDA and Intel MKL (Math Kernel Library). CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia and MKL is used for optimizing mathematic operations.

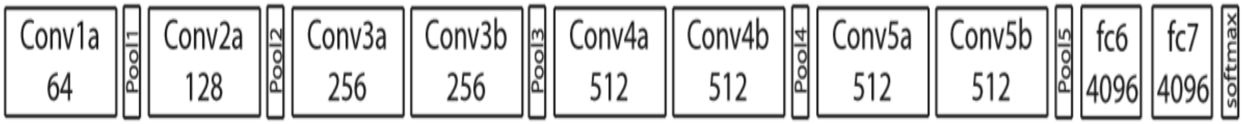

Fig.1 The architecture of the C3D Convolution Neural Network

As shown in above figure, C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are 3×3×3 with stride 1 in both spatial and temporal dimensions. Number of filters is denoted in each box. The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are 2×2×2, except for pool1 is 1×2×2. Each fully connected layer has 4096 output units. [36]

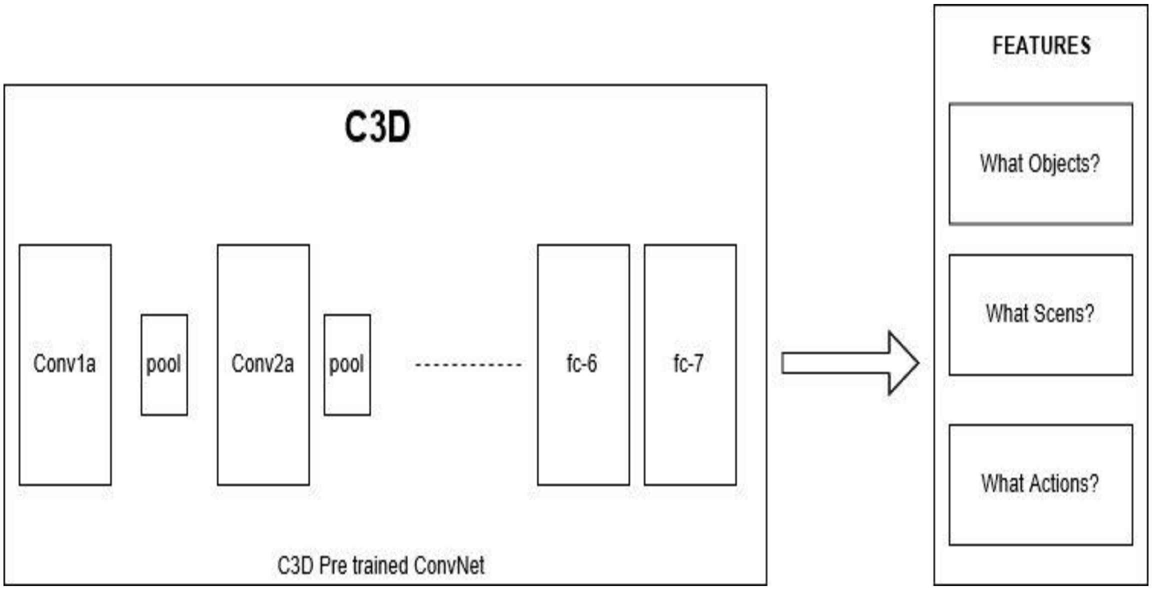

Fig. 2 Extracting features from the C3D ConvNet

3D ConvNets are more suitable for spatiotemporal feature learning compared to traditional 2D ConvNets. A homogeneous architecture with small 3x3x3 convolution kernels in all layers is among the best performing architectures for 3D ConvNets. C3D learns internally. It starts by focusing on appearance in the first few frames and tracks the salient motion in the subsequent frames. For example, first it identifies each person in the first few frames and then starts tracking the motion of each person in the subsequent frames.

3.2.2 Averaging Technique:

For each video, 32 × 4096 feature vector is obtained using the averaging technique and the summary of the technique is shown in Fig. 3

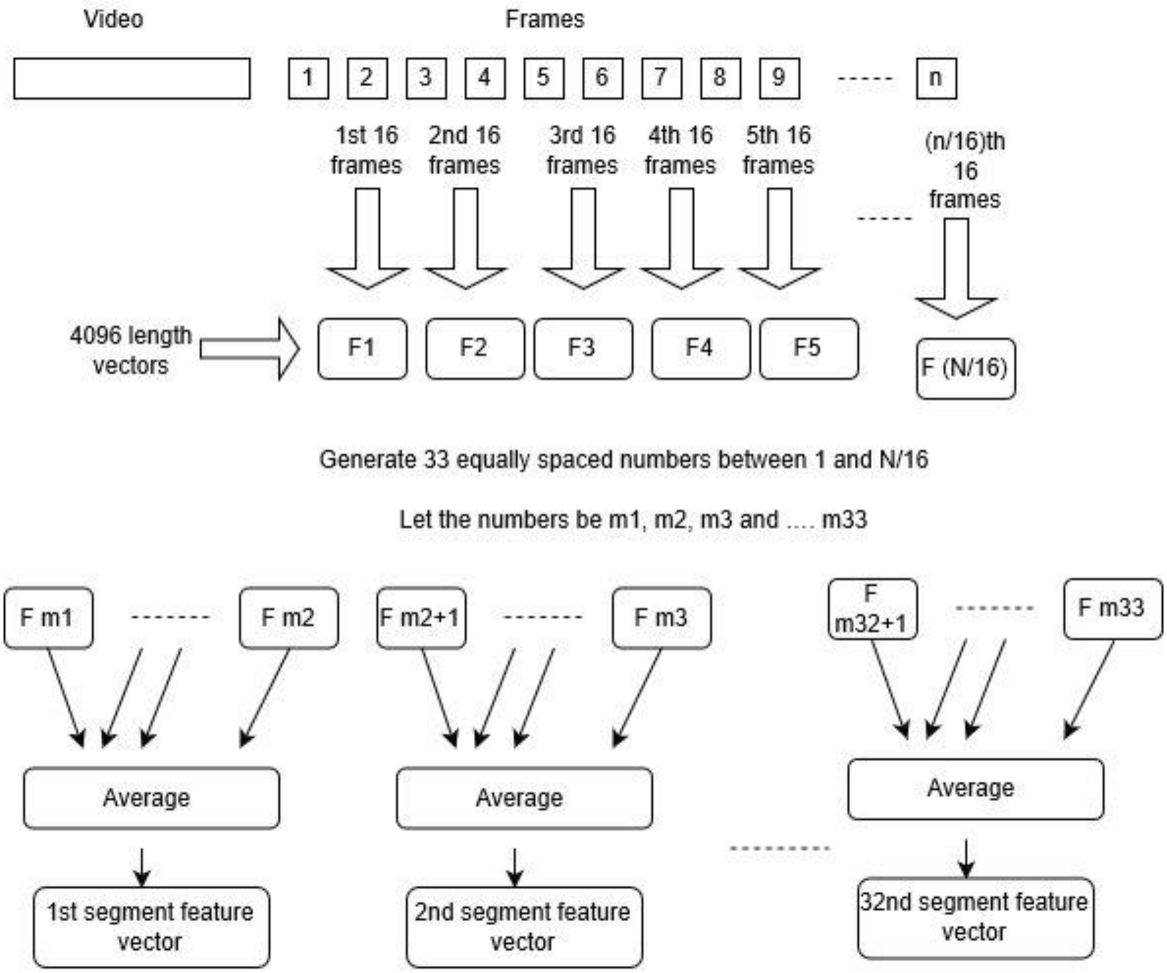

Fig. 3 Summary of the averaging technique

3.3 Model Architecture

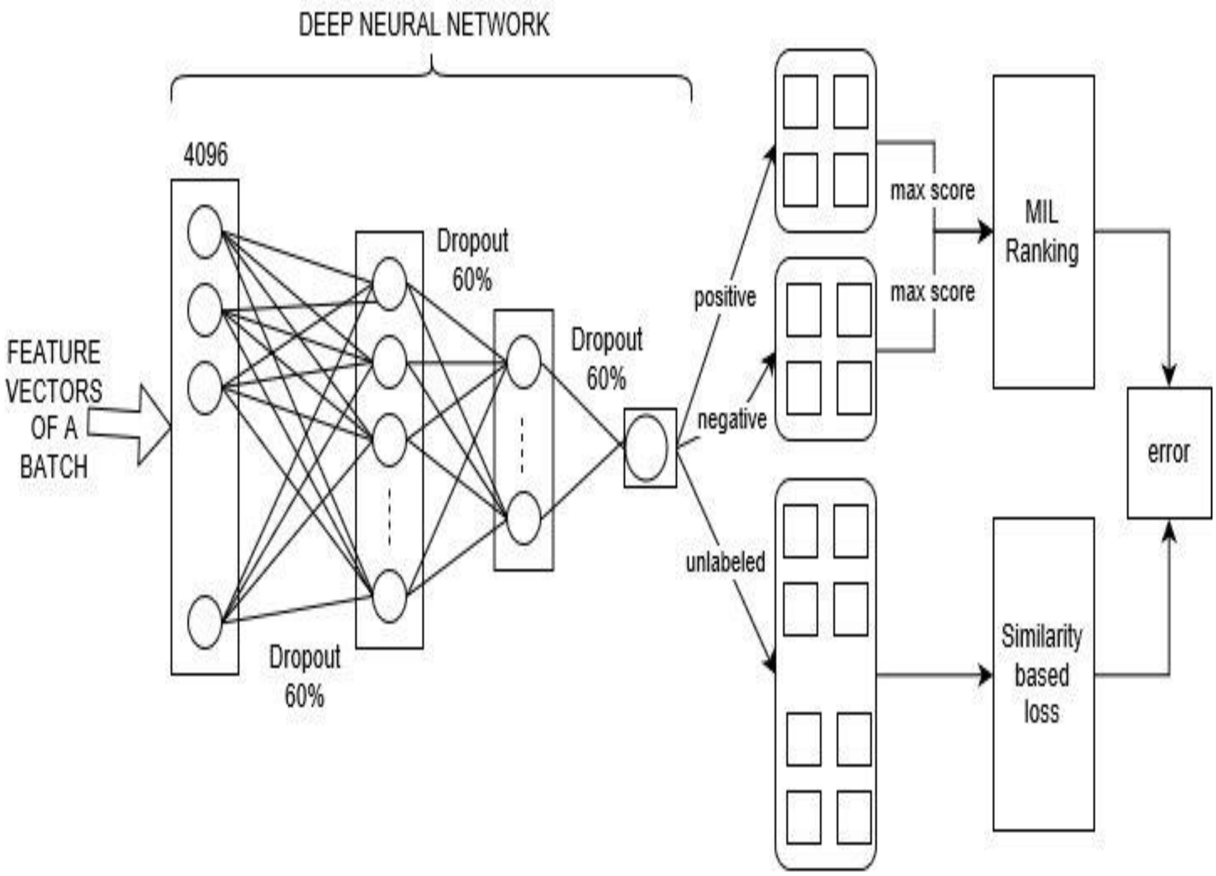

Fig. 4 The architecture of the Deep Neural Network

The above diagram shows the architecture of the deep learning model used for SSMIL along with the feature extraction. The deep neural network consists of 4 layers. Layer 1 consisting of 4096 nodes which are fed with the output of C3D feature extraction model. Layer 2 consists of 512 nodes after a drop out of 60% from layer 1. Layer 3 consists of 32 nodes after a dropout of 60% from the layer 2. Layer 4 consists of only the output node giving the score of each segment of a video after dropout of 60%. The dropout is used to reduce the over fitting of the network.

The training is done iteratively. In each iteration, a batch of videos is selected random and those videos are split in the ratio of 1:2, the labeled part and the unlabeled part respectively. In the labeled part of videos, equal numbers of normal and anomalous videos are present. For the labeled part, the loss is calculated using the MIL ranking loss function and for the unlabeled part, the similarity based loss is computed. Both the losses add up forming the total error of the iteration and this error is back propagated through the deep neural network.

3.4 Similarity based Semi Supervised Multiple Instance Learning (SSMIL):

This is the name given to the approach that leverages the benefits of Multiple Instance learning and Semi – supervised learning.

3.4.1 Multiple Instance learning (MIL):

In standard supervised classification problems using support vector machine, the labels of all positive and negative examples are available and the classifiers learned using the following optimization function [35].

$$\min_{w}[\frac{1}{k}\sum_{i=1}^{k}\max(0,1-y_i(w.\emptyset(x)-b))] \qquad - (1)$$

It is also called as hinge loss function. Here y_i represents the label of each example, $\emptyset(x)$ denotes feature representation of an image patch or a video segment, b is a bias, k is the total number of training examples and w is the classifier that is to be learned. To learn a robust classifier, accurate annotations (labels) of positive and negative examples are needed. In the context of supervised anomaly detection, a classifier needs temporal annotations of each segment in videos. However, obtaining temporal annotations for videos is time consuming and laborious.

MIL relaxes the assumption of having these accurate temporal annotations. It is a type of supervised learning where a set of instances is labeled but not the individual instances. This is helpful for training the model with untrimmed videos, i.e. the label of each training video is known but the individual labels of the segments/sequences of the video are unknown.

In the training data, each video is labeled as either normal or anomalous. In a video labeled normal, it is sure that all the individual segments are normal whereas in a video labeled as anomalous, it is unsure exactly which individual segments in the video are anomalous. In mil definition, each anomalous video is labeled as a positive bag and each normal video is labeled as negative bag. In a positive bag, there exists at least one anomalous segment but in a negative bag, all the segments are normal.

MIL tries to predict the labels either at bag level or at the instance level, i.e. either at video level or instance level. The training data is already at instance level, so the goal is to achieve instance level labels for new videos.

3.4.2 Deep MIL Ranking Model

Anomalous behavior is difficult to define accurately, since it is quite subjective and can vary largely from person to person. Further, it is not obvious how to assign 1/0 labels to anomalies. Moreover, due to the unavailability of sufficient examples of anomaly, anomaly detection is usually treated as low likelihood pattern detection instead of classification problem.

The anomaly detection is posed as a regression problem. It is desired that the anomalous video segments to have higher anomaly scores than the normal segments. The straightforward approach would be to use a ranking loss which encourages high scores for anomalous video segments as compared to normal segments. However, in the absence of video segment level annotations, multiple instance ranking objective function is proposed [30]:

$$\max_{i \in B_a} f\left(V_a^i\right) > \max_{i \in B_n} f\left(V_n^i\right) \qquad - (2)$$

Here $\max$ is taken over all video segments in each bag (video), $f\left(V_a^i\right)$ is the anomaly score of i^{th} segment in an anomalous video and $f\left(V_n^i\right)$ is the anomaly of i^{th} segment in a normal video, B_a and B_n represent the positive and negative bag respectively. Instead of enforcing ranking on every instance of the bag, ranking is enforces on only two instances having the highest anomaly score respectively in the positive and negative bags/videos. The segment corresponding to the highest anomaly score in the positive bag is most likely to be the true positive instance (anomalous segment). The segment corresponding to the highest anomaly score in the negative bag is the one looks most similar to an anomalous segment but actually is a normal instance. This negative instance is considered as a hard instance which may generate a false alarm in anomaly detection. By using the below equation, the positive instances and negative instances are pushed far apart in terms of anomaly score. The ranking loss is therefore given as follows:

$$l = \max\left(0,1 - \underset{i \in B_a}{max} f\left(V_a^i\right) + \underset{i \in B_n}{max} f\left(V_n^i\right)\right) \quad - (3)$$

Here l represents the loss, V_a^i represents the i^{th} segment of an anomalous video and V_n^i represents the i^{th} segment of a normal video.

3.4.3 Semi Supervised learning

There is a very limited training labeled videos in the dataset. Semi-supervised learning is a class of machine learning tasks and techniques that also make use of unlabeled data for training, typically a small amount of labeled data with a large amount of unlabeled data. This learning falls between supervised learning and supervised learning.

Our model learns better with each iteration. In each iteration, some positive videos, some negative videos and some unlabeled videos are taken. For the labeled videos, the Deep MIL ranking model is used, but for the unlabeled videos, similarity-based scoring method is used.

There is always a similarity in video segments. All the robbery sequences will be of similar type and all the explosion sequences will be of similar type. So, the model needs to be known that similar kind of video sequences need to be given anomaly scores very close to each other.

The scores given to such segments depend on the similarity between the segments. More is the similarity between the two segments, closer is the score given to them by the model. For this to be achieved, the following cosine similarity based loss is also added to the Eq. 3 to get the total error of a batch of videos:

$$\sum_{i=1}^{n}\sum_{j=1}^{k} w_{i,k_j}\left(f_i - f_{k_j}\right)^2 \qquad \text{- (4)}$$

Here n is the total number of segments of the unlabeled videos, j = 1 to k represents the k nearest neighbors of the i^{th} segment, f_i is the anomaly score of the i^{th} segment and f_{k_j} represents the scores of the k nearest neighbors, w_{i,k_j} represents the cosine similarity score between i^{th} and k_j segments. Note that each video has 32 segments.

The above similarity-based loss function is used for only the unlabeled videos. For the labeled videos, the deep MIL Ranking model is used.

3.5 Loss Function

The total loss for each iteration is the aggregation of loss due to Deep MIL ranking model and loss due to the semi-supervised learning. So, the aggregated loss equation would be the sum of the Eq. 3 and 4.

$$l = \max\left(0, 1 - \max_{i\in B_a} f\left(V_a^i\right) + \max_{i\in B_n} f\left(V_n^i\right)\right) + \sum_{i=1}^{n}\sum_{j=1}^{5} w_{i,k_j}\left(f_i - f_{k_j}\right)^2 \qquad \text{- (5)}$$

CHAPTER 4

EXPERIMENTAL PROCEDURE

4.1 Dataset Description

The Dataset consists of 1900 surveillance videos consisting of 950 normal and 950 anomalous videos. The fps (frames per second) rate is set to 30.

The anomalous videos consists of 13 different anomalies namely, Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. These anomalies are selected because they have a significant impact on public safety.

The entire dataset is divided into 3:1, training data and the test dataset splits.

4.2 Feature Extraction

Steps involved in feature extraction of a Video:

1. Installing the NVIDIA CUDA library. (It is required for the compilation for the C3D model no matter whether GPU or CPU is being used.
2. Building the C3D model
   a. Specifying suitable configuration in the Makefile.config.
   b. make all
   c. make test
   d. make runtest
3. Preparing the input files from which the features are to be extracted.
4. A video is split into 16 frame long clips with a 8-frame overlap between two consecutive clips. These clips are passed to the C3D network to extract fc6 activations. These clip fc6 activations are averaged to form a 4096-dim video descriptor which is then followed by an L2-normalization.

5. Various parameters like feature extractor input file, c3d pretrained model, gpu id (if using GPU), mini batch size, number of mini batches, output prefix file and feature name need to be passed.
6. From step 3, a feature for every 16 frame clip is obtained.
7. Each video is of around 10 -15 seconds and consists of 30 frames per second. For easier computation, each video is assumed to contain 32 segments.
8. $$m = \frac{number\ of\ frames}{16}$$
9. ‘m’ is the number of features that are obtained for each video. 33 equally spaced numbers are generated from 1 to m and then the averaging technique is applied for each split so that features for 32 segments of the video are obtained.
10. Finally, each video has a 32 × 4096 feature vector.

Implementation Details

i. OS: Ubuntu 18.04

ii. Libraries and Packages: CUDA version 10.0, gcc version 5.0, MKL (Math Kernel Library), OpenCV, Boost (C++ Machine Learning Library), MATLAB and many other mentioned in the documentation of CAFFE.

iii. Execution time: 3 complete weeks for all the 1900 videos with RAM 16 Gb and without GPU.

4.3 Training of Model

Keras sequential model is used to build the Deep neural network as shown in Fig. and the loss function mentioned in Eq.

Number of Iterations: 20000

Training data size: 1610 videos (810 Anomalous and 800 Normal)

Batch size: 60

Backend for Keras: Theano

Keras Optimiser: Adagrad (responsible for the learning rate)

Format to store the output model architecture: JSON (Java Script Object Notation)

Format to store the output model wights: MATLAB readable.

Implementation details:

i. OS: Ubuntu 16.04

ii. Language: Python2

iii. Packages: Theano version 1.04 and Keras 1.0.2

iv. Execution time: 2 complete days with RAM 16 Gb and without GPU.

4.4 Testing

The output model and weights obtained from the Training process are used to predict the segment level scores for the Testing Videos consisting of 140 Anomalous Videos and 150 Normal Videos. The scores of 32 segments of a video are to be stored in a single file.

Format to store the score the output scores: MATLAB readable format.

Implementation Details:

OS: Ubuntu 16.04

Language: Python2

4.5 Evaluation

The segment level scores obtained from the Testing process need to be converted to frame level scores. The temporal annotations (frame level labels) for the Testing Videos are available.

Sample Temporal Annotation file:

1. Abuse028_x264.mp4 Abuse 165 240 -1 -1
2. Arson011_x264.mp4 Arson 150 420 680 1267
3. Normal_Videos_003_x264.mp4 Normal -1 -1 -1 -1
4. Normal_Videos_006_x264.mp4 Normal -1 -1 -1 -1

The first line shows that in the corresponding line, there is only one anomaly occurring between frame numbers 165 and 240. The second line shows that there are two anomalies in the video, one occurring between frame numbers 150 and 420 and the other occurring between frame numbers 680 and 1267. The lines 3 and 4 show that there is no anomaly occurring in the videos, i.e. they are normal videos.

Now, both predicted scores and the ground truth labels of all the frames are known. More is the predicted score of the frame, more is the chance of it being anomalous. Hence, all the predicted scores are sorted in descending order and the True positives are calculated. If a frame higher up in the descending order of scores and its ground truth label is normal, then that frame doesn’t add up to the True Positives.

Similarly the False Positives are also calculated by the sorting the scores of the frames in ascending order and the other metrics of the Confusion matrix, i.e. True Negatives and False Negatives are obtained. With the help of confusion matrix, the True Positive Rate and False Positive Rate.

$$True\ positive\ rate = \frac{True\ positives}{True\ positives + False\ negatives}$$

$$False\ positive\ rate = \frac{False\ positives}{False\ positives + true\ negatives}$$

Different True Positive Rates and False Positive Rates can be calculated as the number of frames being tested are increasing. These help in plotting the ROC (Receiver Operating

Characteristic) curve. AUC, the Area Under the Curve of ROC can also be calculated which seem to be the best performance metric for the Anomaly Detection.

4.6 Live Testing

A Graphical User Interface has been created using the PyQtGUI frame work. The authorities can start using the Interface for predicting anomalies once they are satisfied with the above performance measurement.

The input is a pre-recorded video and the output is a graph showing frame numbers on the X – axis and the corresponding anomaly score of the frame on the Y – axis. The graph is generated in real time as the input frames of the video are loaded.

The output graph of one of the Anomalous Videos is shown in Fig.4. It can be seen that there is no occurrence of an anomaly until the frame 400 (except a slight one), the actual anomaly begins somewhere between 400 and 600 frame number and ends at a frame number close to 1000. If hardware is available, an alarm can be raised whenever the anomaly score is above a threshold, say

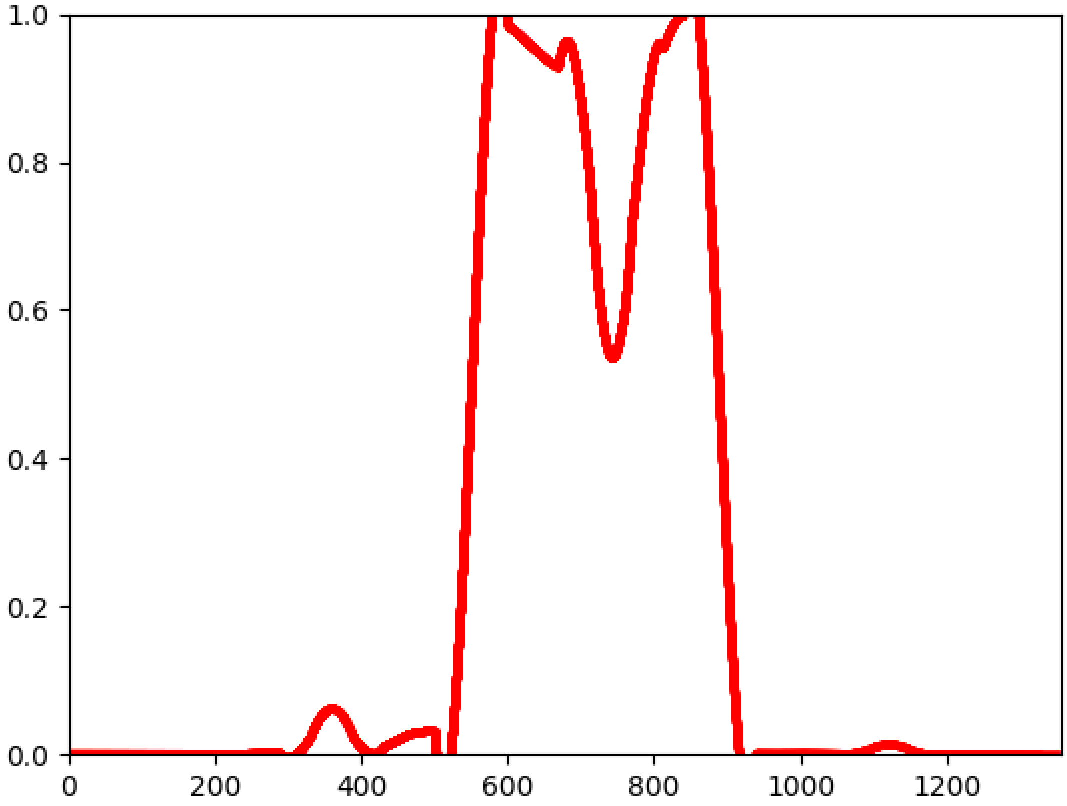

Fig. 5 Output graph of an anomalous video

The output graph of one of the Normal Videos is shown in Fig. 5. It can be seen that there hasn't been a single spike over the range from starting video to ending video of the video. The anomaly scores for the normal videos is almost 0.

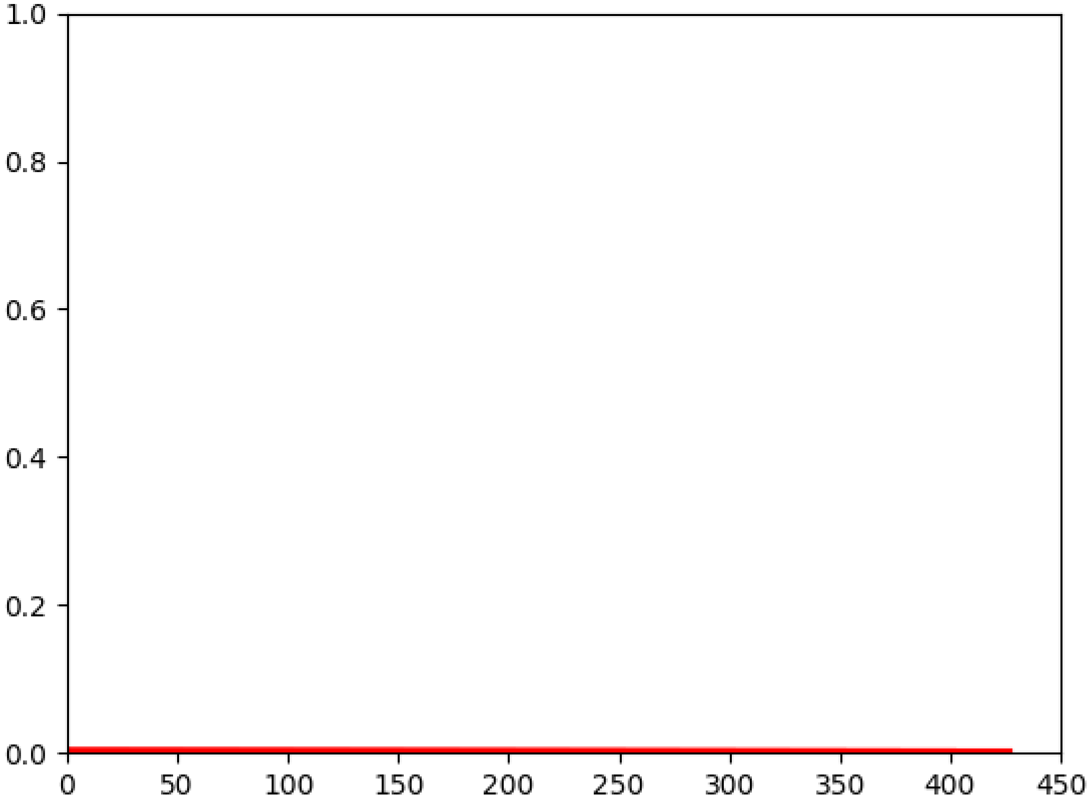

Fig. 6 Output graph of a normal video

4.7 Evaluation of other Approaches

The Supervised MIL models with and without sparsity and smoothness constraints are also trained and evaluated in the same procedure as above and their AUCs are calculated.

For the Rare Label Propagation model, the dataset is generated using the testing videos of the actual dataset. Here, there is no MIL involved. But the input and output is the segment level labels. It has been trained and evaluated with a degree of supervision of 1.5.

The other models, Binary classifier, Hastan and Lu et al are also trained and evaluated using the same dataset.

CHAPTER 5

RESULTS AND DESCRIPTION

MODEL	**AUC (Area Under Curve)**
Binary classifier [37]	50
Hastan [12]	50.6.
Lu et al [10]	65.51.
Deep MIL Ranking Model [30]	75.41.
Rare label propagation with 9600 instances [29]	61
Proposed SSMIL	**79.7.**

Table 1 Comparison of AUCs of different models

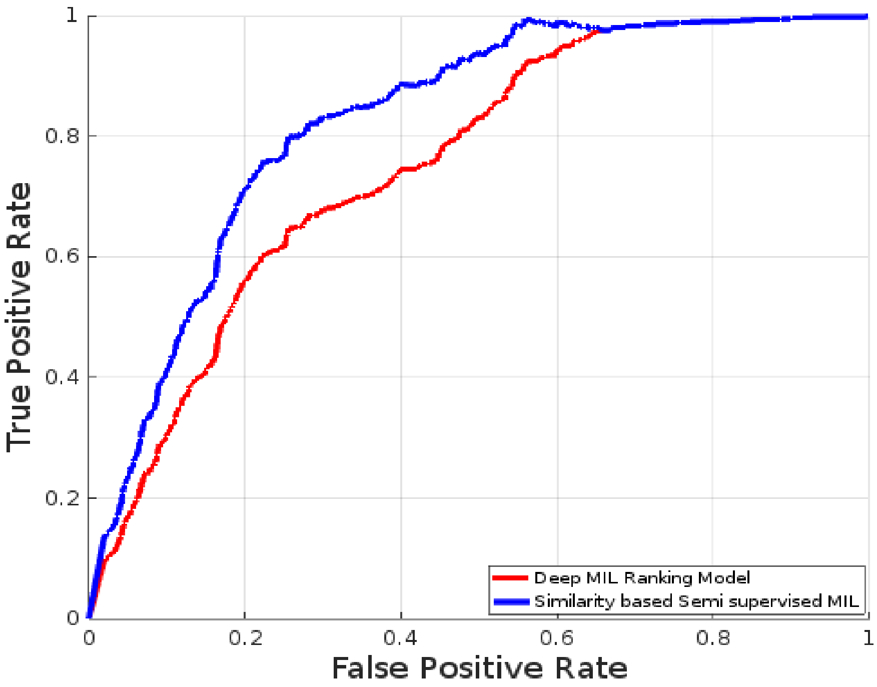

Fig. 7 ROC curves for the Deep MIL model and the Proposed SSMIL

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The proposed SSMIL (Similarity based Semi-supervised Multiple Instance Learning) model has achieved better performance than the existing state of the art approaches with minimum amount of supervision. The AUC measure has been increased by 6% to that of the existing best Anomaly detection technique, i.e Deep MIL ranking model. Also, the time and effort required for the preparation of the training dataset has been significantly reduced than that of the existing approaches.

In Future, we would like to predict the occurrence of an anomaly even before it occurs, Apply the SSMIL model for Anomalous Activity Recognition tasks, i.e. identifying the type of crime (robbery, theft, explosion, e.t.c.) and procure hardware to automatically rise an alarm in the occurrence of anomalous activity in real time.

REFERENCES

[1] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. In BMVC, 2015.

[2] S. Wu, B. E. Moore, and M. Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenesIEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA,2010.

[3] A.Basharat,A.Gritai, and M.Shah. Learningobjectmotion patterns for anomaly detection and improved object detection. IEEE Conference on Computer Vision and Pattern Recognition,Anchorage, AK, USA ,2008.

[4] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas. Abnormal detection using interaction energy potentials. In CVPR, Colorado Springs, CO, USA, 2011.

[5] B.AntiandB.Ommer.Video parsing for abnormality detection. International Conference on Computer Vision. In Barcelona, Spain,2011.

[6] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. IEEE 12th International Conference on Computer Vision, Kyoto, Japan2009.

[7] Y. Zhu, I. M. Nayak, and A. K. Roy-Chowdhury. Context aware activity recognition and anomaly detection in video. In IEEE Journal of Selected Topics in Signal Processing, Pages 91 - 101, 2013.

[8] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. IEEE Transaction on pattern analysis and machine learning ,VOL. 36, NO. 1, 2014.

[9] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In IEEE Conference on Computer Vision and Pattern Recognition,Miami, FL, USA 2009.

[10] C.Lu, J.Shi andJ.Jia. Abnormal event detection at 150 fps in MATLAB. In IEEE International Conference on Computer Vision, Sydney, NSW, Australia,2013.

[11] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA,2011.

[12] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016.

[13] T. Yao, T. Mei, and Y. Rui. Highlight detection with pairwise deep ranking for first-person video summarization. In CVPR, June 2016.

[14]Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury, " Context Aware Activity Recognition and Anomaly Detection in Video", IEEE journal of selected topics in signal processing, vol. 7, no. 1, February 2013.

[15] Weixin Li, Vijay Mahadevan and NunoVasconcelos, "Anomaly Detection and Localization in Crowded Scenes", IEEE transactions on pattern analysis and machine intelligence, vol. 36, no. 1, January 2014.

[16] Dan Xu, RuiSong d, XinyuWu , Nannan Li , Wei Feng and HuihuanQian, " Video anomaly detection based on a hierarchical activity discovery with in spatio-temporal contexts, Elsevier, 2014.

[17] Xuan Mo, Vishal Monga, Raja Balaand Zhigang Fan, " Adaptive Sparse Representations for Video Anomaly Detection", IEEE Transactions on circuits and systems for video technology, vol. 24, no. 4, April 2014.

[18] J.Kooij, M.Liem, J.Krijnders, T.Andringa, and D.Gavrila. Multi-modal human aggression detection. Computer Vision and Image Understanding,Volume 144 Issue C,Pages 106-120, 2016.

[19] S.Mohammadi, A.Perina,H.Kiani, and M.Vittorio. Angry crowds: Detecting violent events in videos. In ECCV, pp 3-18,2016.

[20] A.Karpathy,G.Toderici,S.Shetty,T.Leung,R.Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In CVPR, 2014.

[21] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.

[22] C. Bergeron, J. Zaretzki, C. Breneman, and K. P. Bennett. Multiple instance ranking. In ICML, 2008.

[23] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In CVPR, 2014.

[24] M. Gygli, Y. Song, and L. Cao. Video2gif: Automatic generation of animated gifs from video. In CVPR, June 2016.

[25] A. Sankaranarayanan, S. Alavi and R. Chellappa. Triplet similarity embedding for face verification. arXiv preprint arXiv:1602.03418, 2016.

[26] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. Pattern Recognition, 48(10):2993–1903, 2015.

[27] R. Arandjelovi´c, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In CVPR, 2016.

[28] A. Gordo, J. Almaz´an, J. Revaud, and D. Larlus. Deep image retrieval: Learning global representations for image search. In ECCV, 2016.

[29] Rakesh Pimplikar, Dinesh Garg and Deepesh Bharani. Learning to Propagate Rare Labels. In ACM,New York, Pages 201-210 ,,2014.

[30] Waqas Sultani, Chen Chen and Mubarak Shah. Real-world Anomaly Detection in Surveillance Videos. In CVPR, 2018.

[31] Rui Hou, Chen Chen and Mubarak Shah. Tube Convolutional NeuralNetwork (T-CNN) forActionDetectioninVideos. In CVPR, 2017.

[32] Jefferson Ryan Medel and Andreas Savakis. Anomaly Detection in Video using Predictive Convolutional Long Short-Term memory Networks. In CVPR,New York, 2015.

[33] Yuan-Kai Wan, Ching-Tang Fan, Ke-Yu Cheng, and Peter Shaohua Deng. Real-Time Camera Anomaly Detection For Real-World Video Surveillance. In ICMC,Guilin, China, 2011.

[34] Snehal Anwekar, Isha Walimbe, Priyanka Suryagan,Alisha Gujarathi and Kalpana Thakre. Flexible Content Based Video Surveillance System for crime Prevention based on moving object detection. (IJCSIT) International Journal of Computer Science and Information Technologies,Pages 655-660, Pune, Maharashtra, 2016..

[35] Chengcui Zhang, Xin Chen, Min Chen, Shu-Ching Chen, and Mei-Ling Shyu. A Multiple Instance Learning Approach For Content Based Image Retrieval Using One-Class Support Vector Machine,IEEE International Conference on Multimedia and Expo, Amsterdam, Netherlands,2005.

[36] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani and Manohar . C3D:GenericFeaturesforVideoAnalysis. Dec, 2014.

[37] Stuart Andrews, Ioannis Tsochantaridis and Thomas Hofmann. Support Vector Machines for Multiple-Instance Learning. NIPS'02 Proceedings of the 15th International Conference on Neural Information Processing Systems, Pages 577-584 MIT Press Cambridge, MA, 2003.

APPENDIX

- **Feature Extraction**

  a. Python scripts have to be written to automate the process of extracting the features of a set of videos rather than individually extracting features for each video.

  b. Each frame of the video needs to be resized to 240 × 320 pixels and fix the frame rate to 30 frames per second.

- **Example to extract the C3D features for a video:**

  a. Prepare the input file in the following format:

  <file_path> <starting_frame> <label>: file_path is the path to the video, starting frame is the frame number from where you want to start the feature extraction and end at staring frame number + 16, i.e. feature vector of 4096 length will be generated for the frames starting with <starting_frame> and ending at <starting_frame + 16>. <label> will be either 0 or 1. 0 if extracting features and 1 if fine tuning the C3D model.

  Ex:

  Abuse030_x264.mp4 0 0

  Abuse030_x264.mp4 16 0

  b. Prepare the output file:

  <output_prefix>: Features would be saved with file name starting with <output_prefix>. The layer name would be appended to the file name. For each line in the input file, there will be a output generated which will be stored in file name mentioned in the output file at the corresponding line number.

  Ex:

  Abuse030_x264.mp4/0

Abuse030_x264.mp4/16

c. **"GLOG-logtosterr build/tools/extract_image_features.bin protxt/c3d-sport1m-feature-extractor-frm.prototxt conv3d-deepnetA-sport1m-iter-1900000 0 50 1 protxt/output-list-prefix.txt fc6-1"** is the command used to extract the features and save in the output file. "conv3d-deepnetA-sport1m-iter-1900000" is a pretrained C3D model. And the rest are the arguments. GLOG-logtosterr is to set to 1. Note that the product of the batch size and the number of minibatch should be equal to the number of frames of the video for which the features are being extracted. The batch size depends on the computational power on the machine.

d. FFMPEG (Fast Forwards Moving Pictures Expert Group) is a command line tool to calculate the total number of frames in a video.

e. The output saved would be in MATLAB readable format.

f. They need to be loaded in a MATLAB script and then the averaging technique is to be applied to condense the features to 32 segments with the help of 'linspace' function.

g. In this way 32 × 4096 real valued vector length feature is obtained. The above steps are to be repeated for all the videos.

- C3D User Gude:

  a. https://docs.google.com/document/d/1-QqZ3JHd76JfimY4QKqOojcEaf5g3JS0lNh-FHTxLag/edit

  b. https://github.com/facebook/C3D

- Dataset link: https://visionlab.uncc.edu/download/summary/60-data/477-ucf-anomaly-detection-dataset