Minor Project Report

on

“Anomaly Detection in Surveillance Videos using Deep Learning”

Jay Bangar (180001022), Aditi (180001002), Sai Aravind (180001063)

Problem Description

Surveillance cameras are increasingly being used in public places like streets, intersections, banks, shopping malls, etc. to increase public safety. Still, the monitoring capability of law enforcement agencies has not kept pace. The result is that there is a glaring deficiency in the utilization of surveillance cameras and an unworkable ratio of cameras to human monitors. One important task in video surveillance is detecting anomalous events such as traffic accidents, crimes or illegal activities. Generally, anomalous events occur less frequently as compared to normal activities. Therefore, development of intelligent computer vision algorithms for automatic video anomaly detection will reduce the waste of labor and time. The goal of our practical anomaly detection system is to timely signal an activity that deviates from normal patterns. Our goal is to make an intelligent neural network such that it can distinguish between normal and anomaly videos i.e. it can **classify** the videos in the above-mentioned two categories.

We present a model which learns anomalies through a deep MIL (Multiple Instance Learning) framework by treating normal and anomalous surveillance videos as bags and short segments/clips of each video as instances in a bag. Based on training videos, the network automatically learns an anomaly ranking model that predicts high anomaly scores for anomalous segments in a video. And during testing, a long untrimmed video is divided into segments and fed into our deep network which assigns an anomaly score for each video segment such that an anomaly can be detected.

Analysis and Design

Data Collection and Preprocessing:-

We are using the **UCF Crime Dataset**. It can be accessed from this link : **UCF Crime Dataset** .The dataset consists of more than 1700 long and untrimmed surveillance videos with combined duration of 128 hours and total size of 96 GB. It has videos of 13 realworld anomalies, including *Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism*. The above selected anomalies are used because they have a significant impact on public safety.

Annotation : Since we do binary classification, we just need video-level labels for training the data set.

Preprocessing : Our model needs input in the form of features of a video. For this first the video is divided into a fixed number of non overlapping segments (32) and the features are extracted for those segments. These segments constitute a bag in the MIL ranking model. **Facebook's C3Dv1.0** with default setting is used for feature extraction. Before computing features each video frame is resized to 240 x 360 pixels and the frame rate is set to 30 fps. To compute the features of a video segment, an average of all 16 frame clip features within that segment is taken.

Dataset we used: As the size of the original dataset was too large and we were unable to download the complete dataset doing feature extraction on our own was not possible. So we used an already existing preprocessed data set. The data set can be accessed from the given link : **Dropbox Link** **.** The data in the link is already processed through the C3D feature extraction is of low size too (around 3.5 GB). It consists of a total 1786 videos, out of which there are **900 Anomaly Videos and 886 Normal Videos.**

Training and Testing Split : We used a total 1546 videos for training our network (780 Anomalous Videos and 766 Normal Videos). The remaining 240 videos (120 of each type) are used for testing the network for accuracy.

Implementation Details:-

Training: We have made a Fully Connected, 3 Layered, Neural Network. The Input layer has 4096 neurons, the first hidden layer has 512 neurons, the second hidden layer consists of 32 neurons and the final layer (output layer) has 1 neuron (since we need only binary classification). We have also used dropout in between the layers for better generalization in training. **ReLU** activation function is used for the first hidden layer and **Sigmoid** function for the output layer.

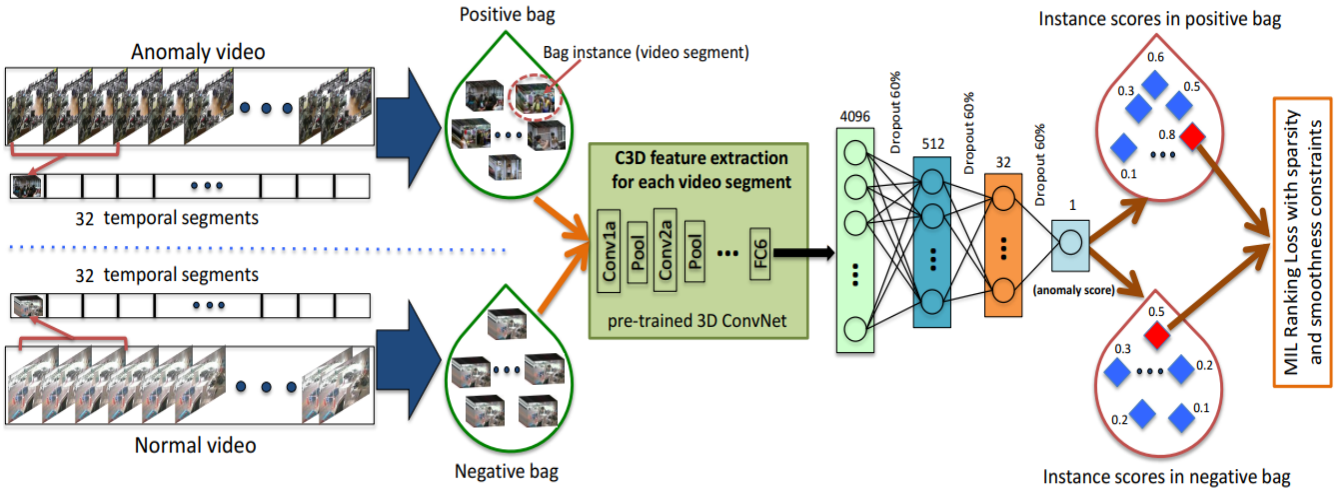


Figure 1. The flow diagram of the proposed anomaly detection approach. Given the positive (containing anomaly somewhere) and negative (containing no anomaly) videos, we divide each of them into multiple temporal video segments. Then, each video is represented as a bag and each temporal segment represents an instance in the bag. After extracting C3D features [36] for video segments, we train a fully connected neural network by utilizing a novel ranking loss function which computes the ranking loss between the highest scored instances (shown in red) in the positive bag and the negative bag.

During training we have used the batch mode for training, i.e. we trained our network on multiple batches of videos, where in each batch there are 50 videos (25 of each type). The optimizer used for training is **Adagrad** (Adaptive Gradient) with an initial learning rate as 0.01 (we vary it for testing purposes). We have used a custom loss function suited for our deep MIL ranking model. The aim is to give higher anomalous scores segments which consist of any anomalous activity. Since our batch size 50 and there are total of 1546 videos in out training dataset, hence on average it takes almost 32 iterations to cover the complete training dataset and we have trained our dataset on different number of iterations but the maximum was 3200 (~100 epochs). Training over 3200 iterations took an average of 5 hours and we had to train on different hyper-parameters, so we decided to keep 3200 as cap of our iteration value. For making and training the model we have used Python's inbuilt libraries.

Testing: At the end of training we aim to classify a video either as anomalous or normal. This classification is based on the anomaly scores, the scores predicted from the network are non binary values (Regression Problem) hence, we check the maximum of the predicted score and if that score is >=0.8 then we classify the video as anomalous, otherwise we put the video in the normal video category.

Link to the implemented code: **Training** and **Testing**

Testing Results

For testing purposes we have used 240 videos out of which 120 are normal and the remaining 120 are anomalous. According to our training, the higher the segment score in a video, the higher the chances are there of that video containing anomalies. So for a single video we store the predictions in a list and then find the maximum of all the elements in the list, if the maximum value is greater than 0.8 (this is the threshold value we used) then the video is classified as an anomalous video.

We have checked the testing accuracy on models trained by varying the value of learning rate (passed as an argument to the adagrad) and dropout value. These were the only 2 places we thought we could change the values.

Information Summary:

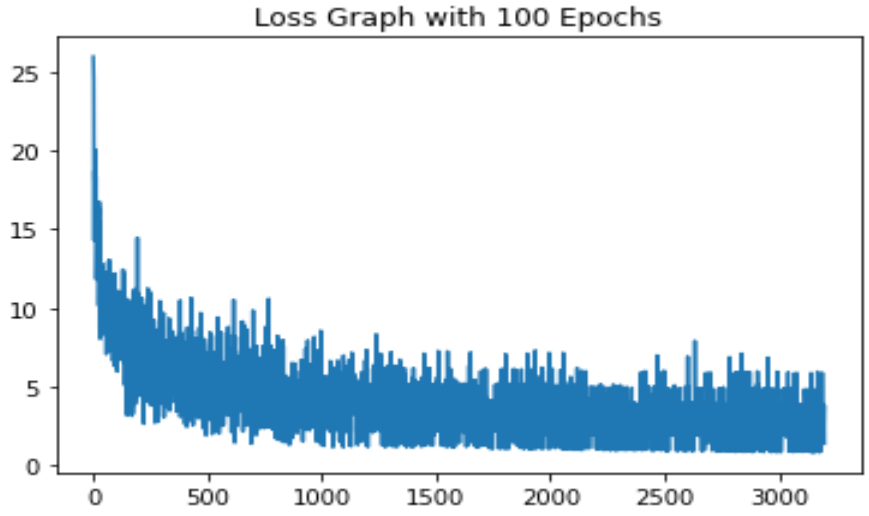
Training : 1546 Videos (780 Anomaly Videos and 766 Normal Videos)
Testing : 240 Videos (120 videos of each)
Batch Size : 50 (25 videos of each type in a batch)
Optimizer : Adagrad (Adaptive Gradient)
Threshold used for classification : 0.8

Result Table:

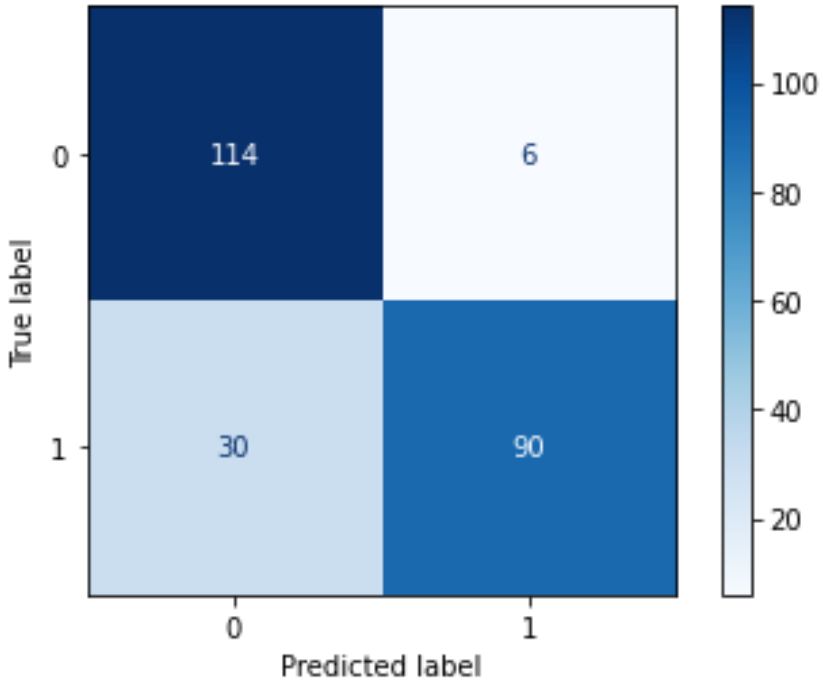
Iterations	**Learning Rate**	**Dropout** (at each layer)	**Training Time**	**Testing Accuracy**
320	0.01	60%	15 Minutes	80.83 %
640	0.01	60%	27 Minutes	83.83 %
1600	0.01	60%	70 Minutes	84.16 %
3200	0.01	60%	240 Minutes	85.00 %
3200	0.03	70%	270 Minutes	83.75 %
3200	0.05	50%	360 Minutes	84.58 %
3200	0.07	60%	320 Minutes	81.25 %
3200	0.09	50%	350 Minutes	81.25 %
3200	0.1	50%	320 Minutes	80.83 %

Case : Initial learning Rate = 0.01 and Dropout = 60%

Loss Graph:



Confusion Matrix :



Observations :

- On increasing the number of iterations (from 320 to 3200), the testing accuracy increases.
- The best results are achieved when learning rate = 0.01 and dropout = 60%.
- Increasing the initial learning rate does not necessarily increase the testing accuracy.

Conclusion:

The implemented anomaly detection model using Deep Learning gives promising results. Better performance might be achieved by increasing the number of iterations with computational support. The binary classification could also be further extended into a 13 - Class Classification problem for each type of anomaly in our dataset.

References :

- Real-world Anomaly Detection in Surveillance Videos