Ramaiah Institute of Technology

(An Autonomous Institute, Affiliated to VTU)

MSR Nagar, MSRIT post, Bangalore-54

A MACHINE LEARNING PROJECT

ON

Anomaly Detection in Human Behaviour using Video Surveillance

Submitted by

Name of the student :	**Neha Sharma**	USN number:	**1MS16CS148**
Name of the student :	**Mohammed Annan**	USN number:	**1MS16CS057**
Name of the student :	**Rohit Kumar**	USN number:	**1MS16CS080**

In Partial fulfillment of 3rd year BE (CSE) Program

Supervised by

Dr. J. Sangeetha

Department of Computer Science & Engineering

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

M.S.RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE-560054

ABSTRACT:

Traditional passive surveillance is proving ineffective as the number of available cameras for an operator often exceeds the operators ability to monitor them. Furthermore, monitoring surveillance cameras requires a focus that operators can only uphold for a short amount of time.Thus in this paper the focus is on solving the problem of anomaly detection in video sequence through semi-supervised techniques. Each video is defined as sequence of frames. The model is trained with goal to minimize the reconstruction error which later on is used to detect anomaly in the test sample videos. The model was trained and tested on most commonly used benchmarking dataset-Avenue[5] dataset. Experiment results confirm that the model detects anomaly in a video with a reasonably good accuracy in presence of some noise in dataset.

Keywords: ***video surveillance, anomaly detection, semi-supervised learning, unusual activity, video processing, abnormal behavior.***

TABLE OF CONTENTS

Sl.NO	**Criteria**	**Page No**
1	**Introduction**	**4**
2	**Literature Survey**	**6**
3	**Problem Formulation**	**12**
4	**Methodology**	**13**
5	**Experiment**	**16**
6	**Conclusion**	**21**
7	**References**	**22**

1. INTRODUCTION

With the increasing number of anti-social activities that have been taking place, security has been given utmost importance off late and it is paramount that every citizen plays his share in warranting the safeguarding of society.

Many organizations have mounted CCTV cameras for the constant monitoring and invigilation of people in public areas and their interactions. For a developed country such as ours, with a population of 1.33 billion, every person is captured by a camera ~ 70 times a day. A lot of video is spawned and stored for a certain time duration. A 704x576 resolution image which is recorded at 25fps will produce roughly 20GB data per day. Since constant monitoring of data by humans to judge if the events are abnormal is a near impossible task, and requires a colossal workforce and their constant attention and awareness, it calls for a need to automate the same. Also, there is a necessity to show which frame and which parts of the video contain the unusual activity which can aid in faster judgment of that anomalous activity being abnormal.

Further, the definition of an anomaly depends on what context is of interest. A video event is termed as an anomaly when something unusual happens in the video frame which does not confer to the usual norms. With rapid growth of video data, there is an increasing need not only for recognition of objects but in particular for detecting the rare or unusual objects too.

Types of anomalies :

- **Point Anomaly**: A lone instance of data is said to be anomalous if it is too far off from the remainder instances. *Business use case :* Detecting credit card fraud based on the amount spent from that card.
- **Contextual Anomaly**: The abnormality in this case is context definitive. This type of anomaly is frequent in time-series data. *Business use case:* Spending $200 on grocery and food every day during the holiday season is typical, but may be irregular otherwise.
- **Collective Anomalies:** A set of data samples/instances simultaneously helps in detecting anomalies. *Business use case:* Someone tries to copy data from a remote machine to a local host unexpectedly, an anomaly that would be flagged as a potential cyber attack.

Meaningful events that are of interest in long video data, such as surveillance footage, often have low probability of anomalies occurring. As such, manual detection of these events is a tedious job that often requires more manpower than what is actually available.

Video data, by itself, is challenging to represent and model due to its high dimensionality, noise and a large variety of interactions. Anomalies are also highly contextual and the definition can be ambiguous. Now, there are copious successful cases where anomaly detection has worked well[1,2,3]. However, these methods work by exploiting labelled data which is infeasible and costly. One must record and classify past events and then train the model. This demands for an approach that is increasingly feasible to implement and doesn’t burden the programmer.

2. LITERATURE SURVEY

This section discusses the diverse research papers that are of consequence to this work and present the underlying features and inferences in them.

Khawaja M. Asim, Iqbal Murtza, and Asifullah Khan in [7] presents a supervised approach for dealing with the problem of detecting anomalies in videos. Taking into account the pixel based approach for identifying anomalies, the authors have used k-mean clustering algorithm, accompanied by a posteriori probability based probabilistic model, and region intersection technique for discovering anomalies.

The algorithm consists of the following steps :

i. Selection of points of interest

ii. Interest points description

iii. Feature vector clustering

iv. Construction of an ensemble of key-points

v. Training and testing of the algorithm

vi. Region junction

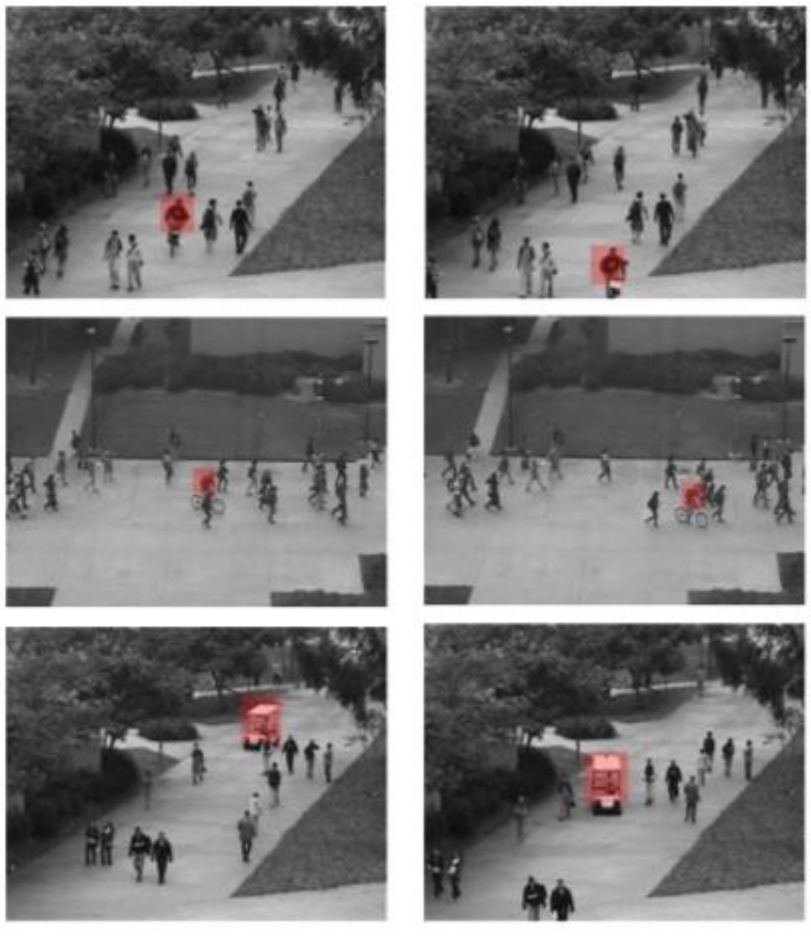


Fig. 1. The presence of a cyclist among pedestrians is detected as an unusual activity, which has a far lower probability of occurrence than the ordinary events, typically pedestrians walking on the pavement.

The technique regards normal events as events having higher probabilities of occurrence. Thickly sampled points are delivered to a probabilistic model through the k-mean clustering to attain the probability of episodes.

The k-mean clustering technique is used for quantization of vectors, and is one of the most prevalent methods for cluster analysis. The algorithms dispenses n sample points in the feature space, randomly selects k points as cluster means, and then allocates every observation to the closest mean. The means of each cluster are updated iteratively.

Let there be n observations, [x1, x2,..,xn], where each xi depicts an observation which is a d-dimensional vector. The clustering algorithm partitions the n observation into a user-defined "k" number of clusters where k<=n. Mathematical representation of the algorithm is shown in Eq 1.

$$\arg_s \min \sum_{i=1}^{k} \sum_{x_j \in s_i} \left\| x_j - \mu_i \right\|^2 \tag{1}$$

A threshold value is applied for differentiating the anomalous events from ordinary ones. The ultimate results of abnormal event detection which are attained from multiple scales are put together through region assimilation. The amalgamation of results of multi-scale unusual event detection using region assimilation helps reduce false positive vigorously. The method is tested on the standard UCSD dataset, detecting anomalies with great success.

Chong and Tay in [1] talk about the most common feature for anomaly detection- video feature representation. Ample research has been done in finding the skillful anomaly detection but finding anomaly in videos is still an open challenge. This is because of large variations of its environment, human continual movement, high space-time complexity and the complex dimensionality of video data. The author also mentions that it is awfully difficult for any anomaly detectors to go with the supervised approach because one will have to train the model for every possible situations and the model will have incredible intricacies. Therefore, the author suggests that the semi-supervised approach could help in learning of the video data.

The author also discusses some semi supervised algorithms and optical flow-based descriptor in contrast to trajectory extraction which requires identifying and tracking of objects, while optical flow methods do not depend upon any such preconditions.

In [2], Sabokrou and Fathy discuss another novel approach for anomaly detection. The authors' proposed method can detect real-time anomalies in crowded scenes. Their work treats a video as a set of non-overlapping cubic patches, and is described using local and global descriptors. These descriptors only find the video properties from different features. Adopting simple and costeffective Gaussian classifiers, the model can distinguish normal activities from the abnormal ones. The work also describes how these local and global features are defined, which is structure similarity between adjacent patches.

The algorithm commences with the creation of a sequence of frames from captured video and the classification of frames as local and global patches using Gaussian distributions after which the final decision is made regarding the nature of those frames. The same has been depicted in fig. 2.

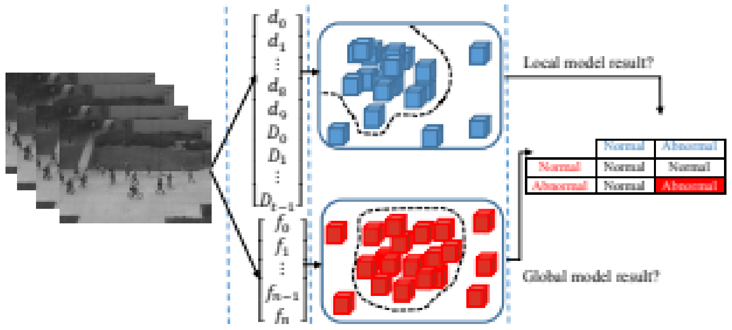


Fig. 2. The scheme of the algorithm: input frames, two views of patches(global and local), modelling the data using Gaussian distributions, and making the final decision.

The author also depicts the detection of anomalies in the UCSD dataset as shown below in fig.3.

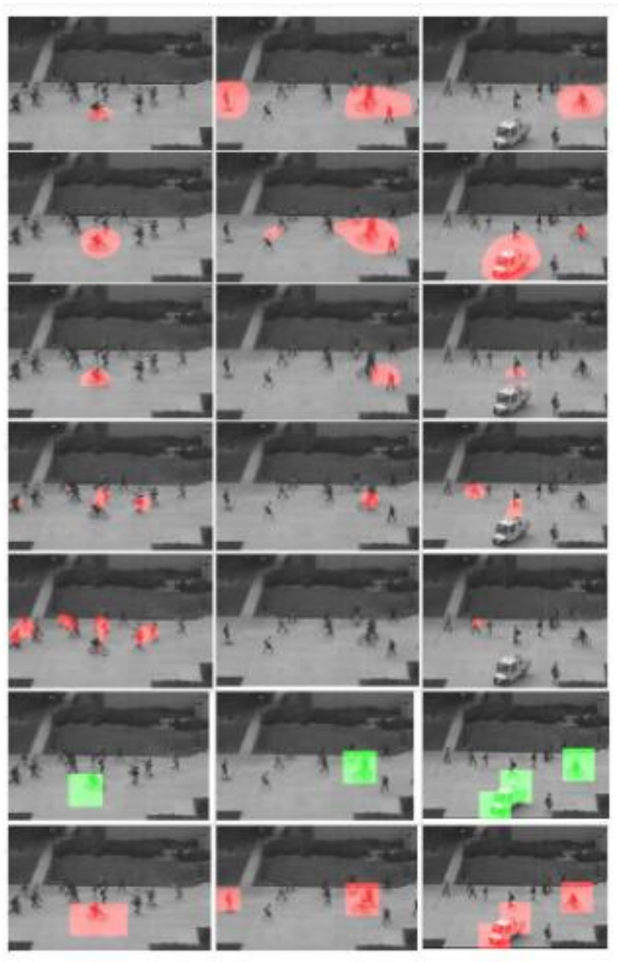


Fig. 3. Example of anomaly detection from three scenes.

In [3], B. Ravi Kiran and Ranjit Parakkal talk about a semi-supervised approach to detecting aberrations in videos.

Semi-supervised learning is a sphere of machine learning which makes use of unlabeled data alongside a small measure of labeled data. It is a blend of supervised and unsupervised learning approaches. Research has shown that a large amount of unlabeled data, when used with a small fraction of labeled data for training, can produce great improvements in the learning accuracies of machine learning models.

The procurement of labeled data for learning problems usually requires a skilled person or an experiment to determine the class labels associated with the instances. The cost linked with the labeling process may thus render a thoroughly labeled training set unattainable. In contrast, the acquisition of unlabeled data is comparably inexpensive. In such situations, a semi-supervised learning can prove to be of great practical significance.

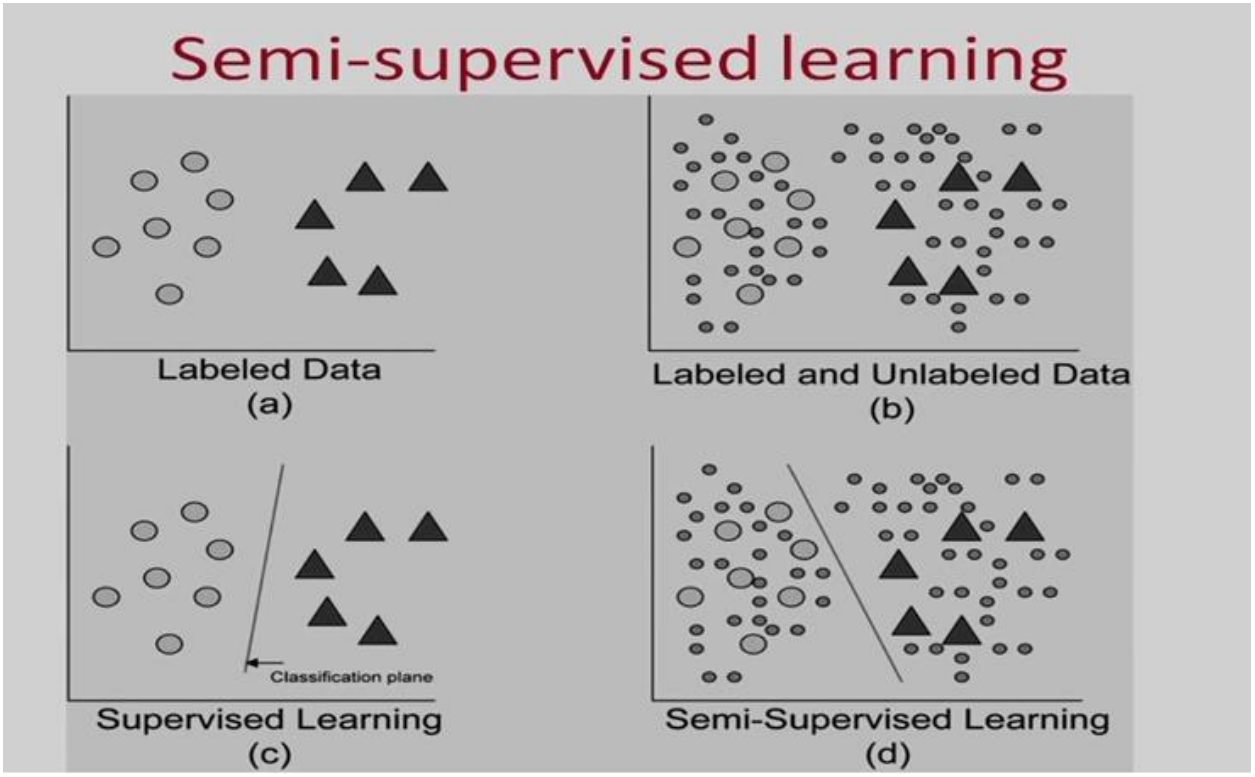


Fig.4. depicts a semi-supervised concept based learning model that binds labeled and unlabeled training instances and makes the most out of both the approaches.

The authors first review the deep convolutional architectures for representation of features along with the generative and predictive models for the task of detecting unusual patterns in the video footages. For representing an unusual activity, the authors make use of an "anomaly-mask" that highlights a suspicious activity in the given frame.

For reconstruction modeling, the paper discusses several dimensionality reduction techniques including :

Principal Component Analysis(PCA) : PCA tries to find the directions of maximal variance in the training dataset. Incase of videos, the model aims to achieve the spatial correspondence between pixel values which are segments of the vector representing a frame at a particular instant of time.
Taking X as the input matrix having a non-zero mean, we find orthogonal projections that disassociate features in the training data, as shown in eq 2 :

$$\min_{W^T W = I} \|X - (XW)W^T\|_F^2 = \|X - \hat{X}\|_F^2 \qquad \text{-(2)}$$

Here, $W^TW = I$ represents an orthogonal reconstructional of the input data matrix X, and the projection XW represents a vector in a lower dimensional space. This reduction in dimensionality captures the anomalous behavior in the samples, since they are not that well reassembled. The Mahalanobis distance between the reconstruction and the original input gives the anomaly score.

Autoencoders : An auto-encoder is an artificial neural network which is used to learn efficient data coding in an unsupervised fashion. They achieve to learn a representation of a set of data for dimensionality reduction by training the mode to ignore useless or unnecessary data, often termed as noise. Now, along with the reduction lateral, a reconstruction lateral is also learnt, wherein the auto-encoder generates a representation closing resembling the original input from the reduced input which is devoid of irrelevant features or dimensions.
These reconstruction based predictive and generative models build representations to minimize the error of reconstruction from the normal distribution in learning models.

In [6], Khalegi and Moin emphasis the use of advanced machine learning deep learning approaches to tackle the problem of detecting anomaly through the video surveillance footage from cameras situated in crowded places. This approach has the advantage of automating feature selection in End-to-end system and also to extract information from high dimension data. The author discusses 3 classes for deep learning based anomaly detection-supervised, unsupervised and semi-supervised. In supervised technique there is a need of labelled data to train the model for the detection but in real world scenario the number of labels will be huge and the dimension of the data too. On the other hand we don’t require any labelled data in case of unsupervised learning technique but the computation turns out to be more complex. Finally, semi-supervised is combines the flavour of both the previous techniques where the model is trained with data which consist of both labelled and unlabelled data.

The method proposed by the author in this paper mainly consists of 2 part – (i)extraction and feature learning (ii)anomaly detection. Apart from the two stages there is also a data pre-processing step which involves background estimation and elimination. Next, the model is made to undergo the training phase using the data volume containing only normal frames. Finally the model is used to detect anomaly from testing videos which consists of abnormality.

Experimental results were based on the UCSD dataset which is considered as benchmark dataset for the purpose of anomaly detection in video and using Mean Absolute Error and Equal Error Rate as major evaluation metrics. The approach showed an increase in accuracy for detecting anomaly.

3. PROBLEM FORMULATION

The following section discusses the problem of detecting unusual activities in frames effectively. The semi-supervised approach is the one that best suits our case. Even though supervised learning methods are the standard, and provide considerably good results, they are just not feasible for large datasets. A camera generates hundreds of gigabytes of video per day, and this video needs to be processed prior to the application of machine learning algorithms. The training dataset requires labelling, and this renders the task extremely time consuming and almost impractical. This leads us towards the unsupervised learning approach, but this method does not provide great results. So we finally stumble upon a hybrid method that takes the best of both worlds and provides the accuracy of supervised models, and the ease of practicality of the unsupervised ones.

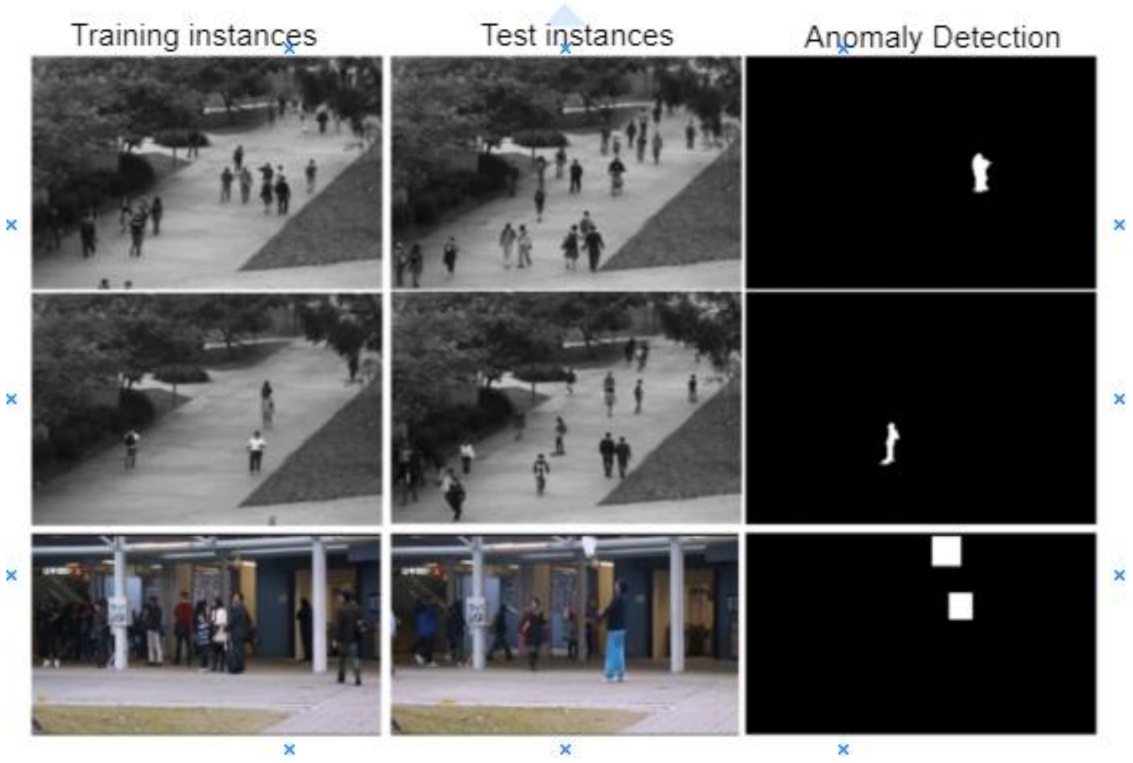


Fig 5. UCSD dataset (top two rows) , portrays the appearance of a cyclist or a skate-boarder on the pavement as an unusual activity. In the Avenue dataset(bottom row), the throwing of papers into the air by a person accounts for an anomaly.

The third row depicts the Avenue dataset where propelling an object in the air is identified as an anomalous event taking place. Since the person in the frame is throwing a bunch of papers, this act is identified by the model as an anomaly.

4. THE METHODOLOGY

The method used in this approach is based on the difference between the older bunch of frames and the most recent ones to detect anomaly in the given video. The model is first trained on the normal videos(without any abnormal activities) with the goal in mind to minimize the reconstruction error between the input video sequence and the output video sequence reconstructed by the trained model with the help of Autoencoder. Once the model is trained the reconstruction error for the normal video is less compared to videos with abnormal events. By setting up a threshold value on the error produced in the testing input, the model is able to detect abnormality in the scene.

Steps involved:

4.1 Data Preprocessing

In this the first step is to convert the input video into frames. Then resize it to 227x227.Next, each frame is normalized by scaling the pixel value between 0 and 1. After that, the images are converted to greyscale to reduce the dimensions. We then clip out negative values if any and finally store them in numpy array for further processes.

4.2 Feature Learning

Feature Learning in general means a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data.To learn regular patterns from training videos, convolutional spatio temporal autoencoder [9] is used.The architecture consists of two parts (i) spatial autoencoder for learning spatial structures of each video frame Fig 6. (ii) temporal encoder-decoder for learning temporal patterns of the encoded spatial structures Fig 7.

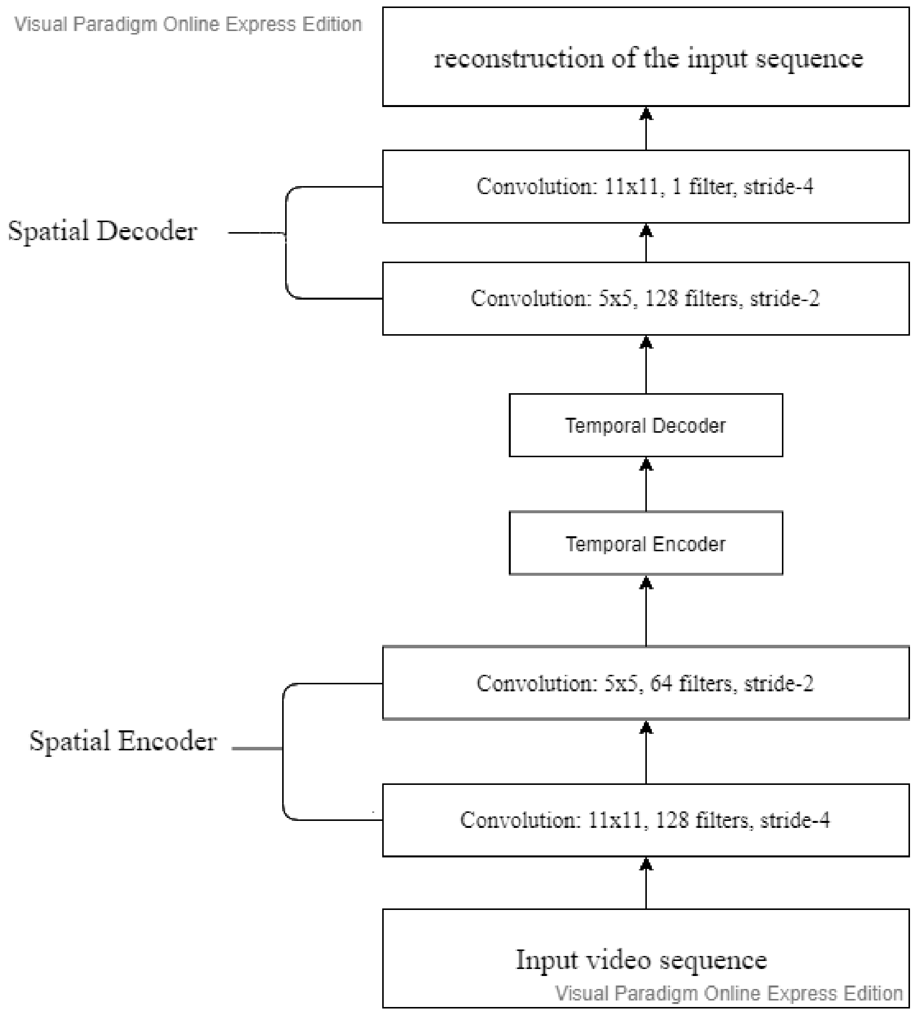

Visual Paradigm Online Express Edition
reconstruction of the input sequence
Spatial Decoder
Convolution: 11x11, 1 filter, stride-4
Convolution: 5x5, 128 filters, stride-2
Temporal Decoder
Temporal Encoder
Spatial Encoder
Convolution: 5x5, 64 filters, stride-2
Convolution: 11x11, 128 filters, stride-4
Input video sequence
Visual Paradigm Online Express Edition

Fig 6.The architecture takes a sequence of length T as input, and output a reconstruction of the input sequence.The dimension is reduced as we go from input and we get back the output with reconstruction value that we tend to decrease in case of training the mod
el using backpropagation algorithm.

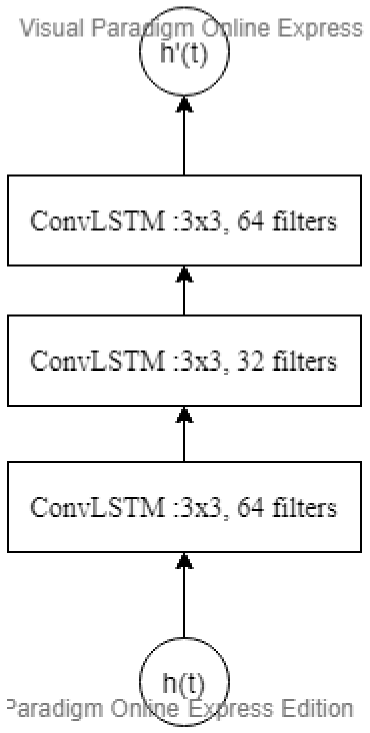

Visual Paradigm Online Express
h'(t)
ConvLSTM :3x3, 64 filters
ConvLSTM :3x3, 32 filters
ConvLSTM :3x3, 64 filters
h(t)
Paradigm Online Express Edition

Fig 7:The zoomed-in architecture of temporal encoder-decoder which comprises of 3 ConvLSTM layers at time t.

4.3 Evaluation Metric:

To evaluate the performance of the model we feed in the testing data and check whether it is capable of detecting anomaly, making sure that false alarm rate is low. By computing the Euclidean distance between input frame and the reconstructed frame , the model flags the video as Anomalous or not depending upon the threshold value that is set during the testing period.

5. EXPERIMENT

5.1 Dataset:

We train our model on most commonly used benchmarking dataset : Avenue [5].In Avenue dataset, there are total 16 training and 21 testing video clips. Each clips duration vary between less than a minute to two minutes long. The normal scenes consist of people walking between staircase and subway entrance, whereas the abnormal events are people running, walking in opposite direction, loitering and etc. The challenges of this dataset include camera shakes and a few outliers in the training data. Also, some normal pattern seldom appears in the training data.

5.2 Model Parameters:

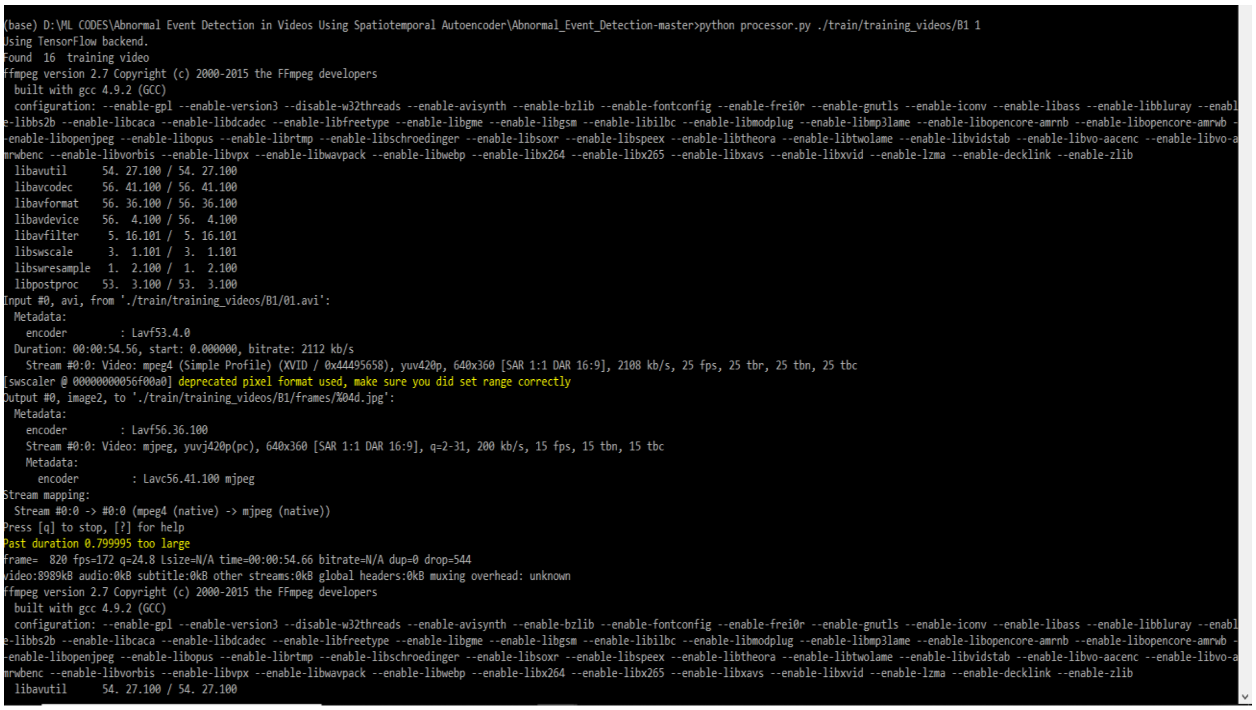
The model was trained by minimizing the reconstruction error of the input volume. Adam optimizer was used in order to set the learning rate automatically based on the model. Mini-batches of size 64 and the model was trained for 30 epochs. Hyperbolic tangent was used as the activation function for spatial encoder-decoder.

5.3 Environment and API's used:

Anaconda was used as the developing environment. Major APIs used in the experiment - numpy, keras, tensorflow, scipy.

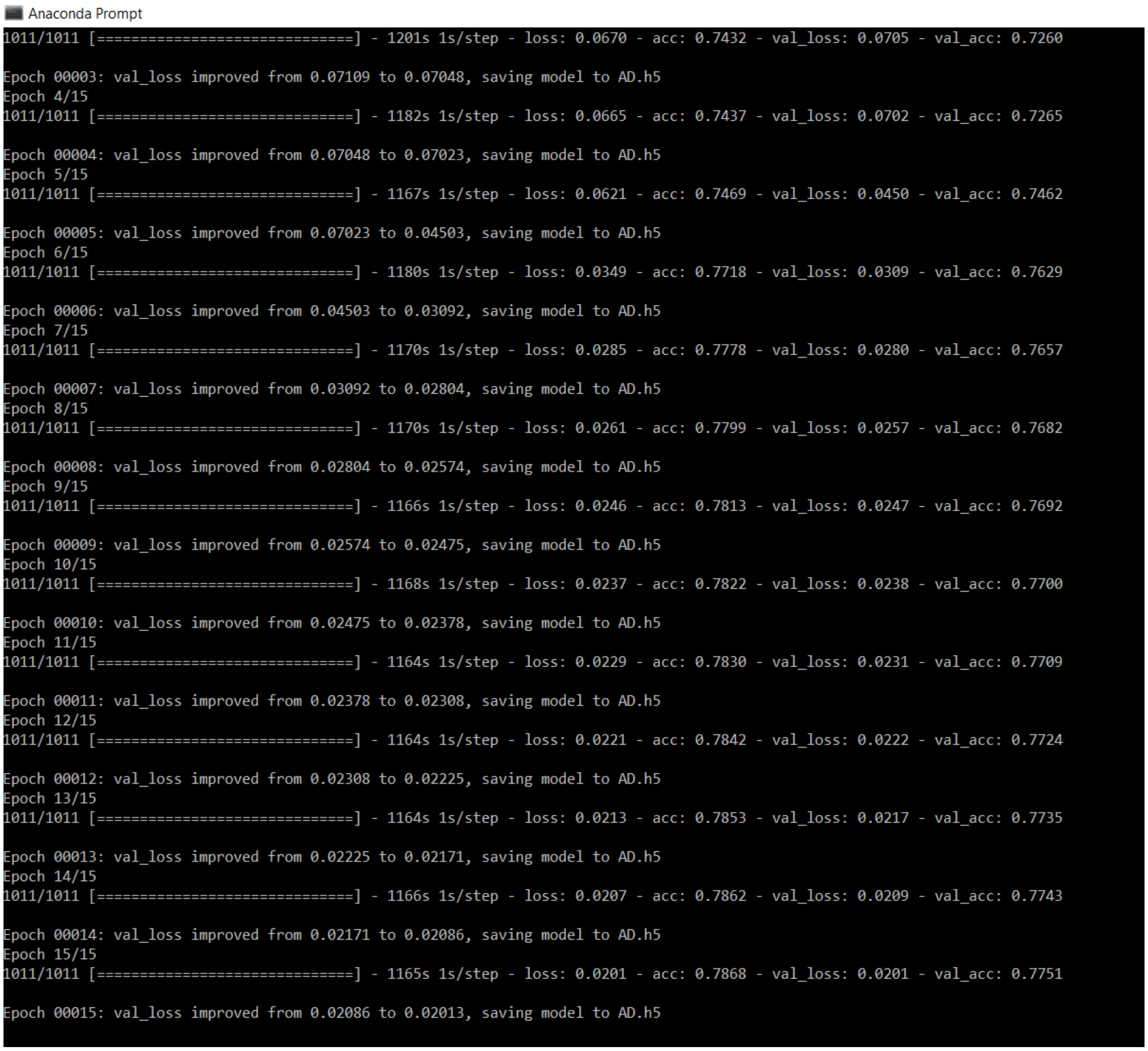
5.4 Experiment Output:

The following fig's shows the snippets of various stages of the experiment. In Fig 3 and Fig 4 the value represents the mean squared error between the input batch and the reconstructed batch from the model.The model detects anomaly and displays the duration(in seconds) that it occurred in the video.



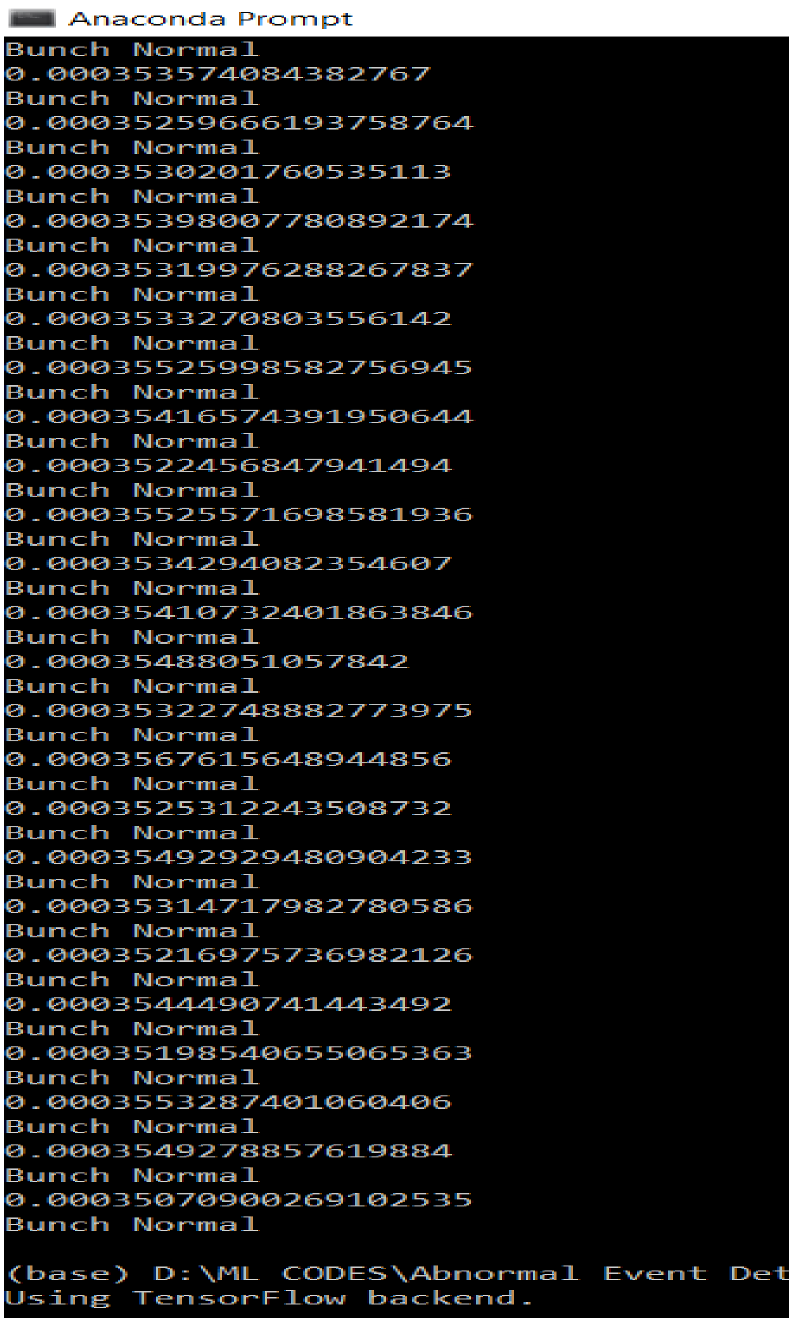
```
(base) D:\ML CODES\Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder\Abnormal_Event_Detection-master>python processor.py ./train/training_videos/B1 1
Using TensorFlow backend.
Found  16  training video
ffmpeg version 2.7 Copyright (c) 2000-2015 the FFmpeg developers
  built with gcc 4.9.2 (GCC)
  configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-avisynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enable-iconv --enable-libass --enable-libbluray --enabl
e-libbs2b --enable-libcaca --enable-libdcadec --enable-libfreetype --enable-libgme --enable-libgsm --enable-libilbc --enable-libmodplug --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb -
-enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libschroedinger --enable-libsoxr --enable-libspeex --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvo-aacenc --enable-libvo-a
mrwbenc --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --enable-lzma --enable-decklink --enable-zlib
  libavutil      54. 27.100 / 54. 27.100
  libavcodec     56. 41.100 / 56. 41.100
  libavformat    56. 36.100 / 56. 36.100
  libavdevice    56.  4.100 / 56.  4.100
  libavfilter     5. 16.101 /  5. 16.101
  libswscale      3.  1.101 /  3.  1.101
  libswresample   1.  2.100 /  1.  2.100
  libpostproc    53.  3.100 / 53.  3.100
Input #0, avi, from './train/training_videos/B1/01.avi':
  Metadata:
    encoder         : Lavf53.4.0
  Duration: 00:00:54.56, start: 0.000000, bitrate: 2112 kb/s
    Stream #0:0: Video: mpeg4 (Simple Profile) (XVID / 0x44495658), yuv420p, 640x360 [SAR 1:1 DAR 16:9], 2108 kb/s, 25 fps, 25 tbr, 25 tbn, 25 tbc
[swscaler @ 00000000056f00a0] deprecated pixel format used, make sure you did set range correctly
Output #0, image2, to './train/training_videos/B1/frames/%04d.jpg':
  Metadata:
    encoder         : Lavf56.36.100
    Stream #0:0: Video: mjpeg, yuvj420p(pc), 640x360 [SAR 1:1 DAR 16:9], q=2-31, 200 kb/s, 15 fps, 15 tbn, 15 tbc
    Metadata:
      encoder         : Lavc56.41.100 mjpeg
Stream mapping:
  Stream #0:0 -> #0:0 (mpeg4 (native) -> mjpeg (native))
Press [q] to stop, [?] for help
Past duration 0.799995 too large
frame=  820 fps=172 q=24.8 Lsize=N/A time=00:00:54.66 bitrate=N/A dup=0 drop=544
video:8989kB audio:0kB subtitle:0kB other streams:0kB global headers:0kB muxing overhead: unknown
ffmpeg version 2.7 Copyright (c) 2000-2015 the FFmpeg developers
  built with gcc 4.9.2 (GCC)
  configuration: --enable-gpl --enable-version3 --disable-w32threads --enable-avisynth --enable-bzlib --enable-fontconfig --enable-frei0r --enable-gnutls --enable-iconv --enable-libass --enable-libbluray --enabl
e-libbs2b --enable-libcaca --enable-libdcadec --enable-libfreetype --enable-libgme --enable-libgsm --enable-libilbc --enable-libmodplug --enable-libmp3lame --enable-libopencore-amrnb --enable-libopencore-amrwb -
-enable-libopenjpeg --enable-libopus --enable-librtmp --enable-libschroedinger --enable-libsoxr --enable-libspeex --enable-libtheora --enable-libtwolame --enable-libvidstab --enable-libvo-aacenc --enable-libvo-a
mrwbenc --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-libx265 --enable-libxavs --enable-libxvid --enable-lzma --enable-decklink --enable-zlib
  libavutil      54. 27.100 / 54. 27.100
```

Fig 8: This snippet displays the data preprocessing part of the experiment.



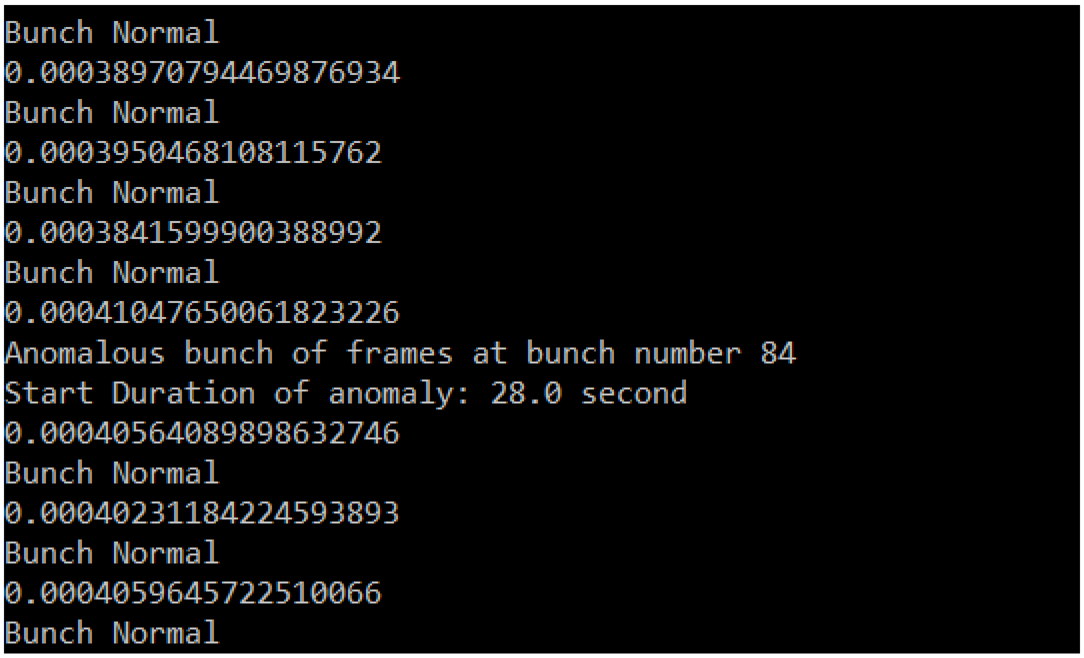
```
1011/1011 [==============================] - 1201s 1s/step - loss: 0.0670 - acc: 0.7432 - val_loss: 0.0705 - val_acc: 0.7260

Epoch 00003: val_loss improved from 0.07109 to 0.07048, saving model to AD.h5
Epoch 4/15
1011/1011 [==============================] - 1182s 1s/step - loss: 0.0665 - acc: 0.7437 - val_loss: 0.0702 - val_acc: 0.7265

Epoch 00004: val_loss improved from 0.07048 to 0.07023, saving model to AD.h5
Epoch 5/15
1011/1011 [==============================] - 1167s 1s/step - loss: 0.0621 - acc: 0.7469 - val_loss: 0.0450 - val_acc: 0.7462

Epoch 00005: val_loss improved from 0.07023 to 0.04503, saving model to AD.h5
Epoch 6/15
1011/1011 [==============================] - 1180s 1s/step - loss: 0.0349 - acc: 0.7718 - val_loss: 0.0309 - val_acc: 0.7629

Epoch 00006: val_loss improved from 0.04503 to 0.03092, saving model to AD.h5
Epoch 7/15
1011/1011 [==============================] - 1170s 1s/step - loss: 0.0285 - acc: 0.7778 - val_loss: 0.0280 - val_acc: 0.7657

Epoch 00007: val_loss improved from 0.03092 to 0.02804, saving model to AD.h5
Epoch 8/15
1011/1011 [==============================] - 1170s 1s/step - loss: 0.0261 - acc: 0.7799 - val_loss: 0.0257 - val_acc: 0.7682

Epoch 00008: val_loss improved from 0.02804 to 0.02574, saving model to AD.h5
Epoch 9/15
1011/1011 [==============================] - 1166s 1s/step - loss: 0.0246 - acc: 0.7813 - val_loss: 0.0247 - val_acc: 0.7692

Epoch 00009: val_loss improved from 0.02574 to 0.02475, saving model to AD.h5
Epoch 10/15
1011/1011 [==============================] - 1168s 1s/step - loss: 0.0237 - acc: 0.7822 - val_loss: 0.0238 - val_acc: 0.7700

Epoch 00010: val_loss improved from 0.02475 to 0.02378, saving model to AD.h5
Epoch 11/15
1011/1011 [==============================] - 1164s 1s/step - loss: 0.0229 - acc: 0.7830 - val_loss: 0.0231 - val_acc: 0.7709

Epoch 00011: val_loss improved from 0.02378 to 0.02308, saving model to AD.h5
Epoch 12/15
1011/1011 [==============================] - 1164s 1s/step - loss: 0.0221 - acc: 0.7842 - val_loss: 0.0222 - val_acc: 0.7724

Epoch 00012: val_loss improved from 0.02308 to 0.02225, saving model to AD.h5
Epoch 13/15
1011/1011 [==============================] - 1164s 1s/step - loss: 0.0213 - acc: 0.7853 - val_loss: 0.0217 - val_acc: 0.7735

Epoch 00013: val_loss improved from 0.02225 to 0.02171, saving model to AD.h5
Epoch 14/15
1011/1011 [==============================] - 1166s 1s/step - loss: 0.0207 - acc: 0.7862 - val_loss: 0.0209 - val_acc: 0.7743

Epoch 00014: val_loss improved from 0.02171 to 0.02086, saving model to AD.h5
Epoch 15/15
1011/1011 [==============================] - 1165s 1s/step - loss: 0.0201 - acc: 0.7868 - val_loss: 0.0201 - val_acc: 0.7751

Epoch 00015: val_loss improved from 0.02086 to 0.02013, saving model to AD.h5
```

Fig 9: Snippet from the training phase of the experiment.



```
Anaconda Prompt
Bunch Normal
0.00035357408438276 7
Bunch Normal
0.0003525966619375876 4
Bunch Normal
0.0003530201760535113
Bunch Normal
0.0003539800778089217 4
Bunch Normal
0.0003531997628826783 7
Bunch Normal
0.000353327080355614 2
Bunch Normal
0.0003552599858275694 5
Bunch Normal
0.0003541657439195064 4
Bunch Normal
0.000352245684794149 4
Bunch Normal
0.0003552557169858193 6
Bunch Normal
0.000353429408235460 7
Bunch Normal
0.0003541073240186384 6
Bunch Normal
0.000354880510578 42
Bunch Normal
0.0003532274888277397 5
Bunch Normal
0.000356761564894485 6
Bunch Normal
0.000352531224350873 2
Bunch Normal
0.0003549292948090423 3
Bunch Normal
0.0003531471798278058 6
Bunch Normal
0.0003521697573698212 6
Bunch Normal
0.000354449074144349 2
Bunch Normal
0.0003519854065506536 3
Bunch Normal
0.000355328740106040 6
Bunch Normal
0.000354927885761988 4
Bunch Normal
0.0003507090026910253 5
Bunch Normal

(base) D:\ML CODES\Abnormal Event Det
Using TensorFlow backend.
```

Fig 10: Snippet from the Testing phase when the video had no anomaly.



```
Bunch Normal
0.00038970794469876934
Bunch Normal
0.0003950468108115762
Bunch Normal
0.0003841599900388992
Bunch Normal
0.00041047650061823226
Anomalous bunch of frames at bunch number 84
Start Duration of anomaly: 28.0 second
0.00040564089898632746
Bunch Normal
0.00040231184224593893
Bunch Normal
0.0004059645722510066
Bunch Normal
```

Fig 11: Snippet from the Testing phase when the video had anomaly.

5.5 Result

The data was trained on a GTX 1050 GPU for 30 epochs. Each epoch took an ETA of 30 min approximately. Sequential model from Keras' API was used to train and test on the Avenue dataset. The model detected anomaly from an anomalous video successfully with an accuracy of 0.77.

6. CONCLUSION

In this paper we have successfully applied deep learning approach to tackle the problem of video anomaly detection over the Avenue dataset. A spatial feature extractor and temporal sequencer ConvLSTM was used to solve this problem. The ConvLSTM layer not only preserves the advantages of FC-LSTM but is also suitable for spatiotemporal data due to its inherent convolutional structure.For the experiment Keras' predefined layers were used. By incorporating convolutional feature extractor in both spatial and temporal space into the encoding-decoding structure, we build an end-to-end trainable model for video anomaly detection.Although the model is able to detect anomaly from the benchmark dataset, real world scenarios could be more complex and thus there are chances of false alarm. Future work could be done on reducing the false alarm in case of complex environment.

7 <u>REFERENCES</u>

[1]. Chong, Yong Shean, and Yong Haur Tay. "Modeling representation of videos for anomaly detection using deep learning: A review." *arXiv preprint arXiv:1505.00523* (2015).

[2]. Sabokrou, Mohammad, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. "Real-time anomaly detection and localization in crowded scenes." In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 56-62. 2015.

[3]. Kiran, B., Dilip Thomas, and Ranjith Parakkal. "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos." *Journal of Imaging*4, no. 2 (2018): 36.

[4] Mahadevan, Vijay, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. "Anomaly detection in crowded scenes." In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1975-1981. IEEE, 2010.

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

[6]. Khaleghi, Ali, and Mohammad Shahram Moin. "Improved anomaly detection in surveillance videos based on a deep learning method." In *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*, pp. 73-81. IEEE, 2018.

[7] Asim, Khawaja M., Iqbal Murtza, Asifullah Khan, and Naeem Akhtar. "Efficient and supervised anomalous event detection in videos for surveillance purposes." In *2014 12th International Conference on Frontiers of Information Technology*, pp. 298-302. IEEE, 2014.

[8] .Taylor, Graham W., Rob Fergus, Yann LeCun, and Christoph Bregler. "Convolutional learning of spatio-temporal features." In *European conference on computer vision*, pp. 140-153. Springer, Berlin, Heidelberg, 2010.