Smart Surveillance System

Mihir Shah
Student, Department of Information Technology
Ramrao Adik Institute of Technology University
DY Patil deemed to be University
Maharashtra , India

Mayur Agre
Student, Department of Information Technology
Ramrao Adik Institute of Technology University
DY Patil deemed to be University
Maharashtra, India

Ashok Chawdhary
Student, Department of Information Technology
Ramrao Adik Institute of Technology University
DY Patil deemed to be University
Maharashtra , India

Ms. Jyoti Deone
Asst. Prof, Department of Information Technology
Ramrao Adik Institute of Technology University
DY Patil deemed to be University
Maharashtra, India

Abstract— Automated Teller Machine (ATM) is one of the major standard and feasible methods for making any financial transactions and is continuously increasing due to its convenience. But ATMs are always at a threat of being robbed. Even if the CCTV records the robbers, they cover their faces with objects like masks when entering the ATM or trying to destroy the machine. Hence, they will be difficult to identify. Thus, there is an essential need to provide high security. But nowadays most surveillance cameras are 'dumb' and cannot analyse the incident in real-time. In this paper, we have tried to build a system for smart video surveillance in the Closed-Circuit Television (CCTV) of ATM and to detect, identify and report any type of potentially suspicious/abnormal/anomalous event that the system might identify. The proposed system involves a deep neural network model for significant detection accuracy. A Convolutional Neural Network (CNN) extracts features from individual frames of a video which are then accumulated using a variant of Long Short-Term Memory (LSTM) that uses convolutional gates. CNN and LSTM are together used for the analysis of local motion in a video.

Keywords— ATM, surveillance, CNN, LSTM.

I. INTRODUCTION

Nowadays, ATM's have facilitated the process of financial transactions such as cash withdrawals, transfers, etc. But it has also become an attraction for thieves and robbers. They see it as an easy place to rob. One of the biggest robberies happened in 2012-2013 when nearly $45 million were stolen by a group of criminals operating in more than 24 countries. This is because, for security, only a few security guards are posted, and a CCTV is installed. These surveillance cameras in ATM centres are only for post incidental manual analysis [1]. If any malicious activity takes place, it will be known only if a person who is present at that moment calls the police. Then police will start investigating the event by manually analysing the CCTV footage. Sometimes thieves will cover the CCTV so it cannot record or destroy the camera that renders the CCTV non-functional. Due to these reasons, it is very important to develop a solution to stop these incidents. This will not only alert the police in time but also help in preventing these incidents and reduce human labour [2].

Earlier surveillance systems were more dependent on human operators. Nowadays because of better efficiencies and reliability, automated systems are preferred and by the security, it is very helpful to detect violence. Applications of deep neural networks include computer vision, image classification, and object detection. Learning Spatio-temporal features of the video feed is essential [3]. For this project, we will use a Spatio-temporal auto-encoder based on a 3D convolution network. The spatial and temporal information is extracted by the encoder, and reconstruction of the frames is handled by the decoder. Identification of abnormal events is done by computing the reconstruction errors [4]. Once identified, an alert notification is sent to the email address registered and the video clip is sent on a dashboard. Thus, there will be a high probability of the thieves getting caught and the incident may be avoided.

II. LITERATURE SURVEY

Prof. Ali Khaleghi and Prof. Mohammad Shahram Moin in their paper [5] has developed a system that detects normal and abnormal video. The first step is the data preparation step. Here, the input video is divided into frames and the next preprocessing step removes the background. The feature extraction step is performed manually or automatic that forms the behavioural structure of the data that is modelled, and the feature representation is obtained. Later the objects are detected using CNN and the final decision is made by a two-class based classifier. They have achieved a lower "equal error rate" than the other compared methods.

In the paper [6], the authors have used a Faster R-CNN

network which consists of a region proposal network and a detector. The RPN proposes several proposals which will be checked by a detector for the presence of objects.

In the paper [7], the authors have used a block RPCA for segmenting foregrounds and backgrounds. Then they use a ConvLSTM for recording the spatial and temporal information. The output is estimated using a weighted Euclidean loss technique that pays more attention to foreground regions than background regions. The regularity score and an experimentally defined threshold. is used for classifying the frame as abnormal/normal. Their model outperformed other state-of-the-art techniques like Conv-AE, ConvLSTM-AE, etc.

The paper [8] compares some of the notable approaches to object detection based on DCNNs based on approach, features, pooling, classifier, etc. Finally, they talk about generalizing the objection detection model and the efficacy of object detection frameworks.

In the paper [9], the main steps are feature extraction, feature description and creation of the model. They have used a CNN with 9 layers. They have used gradient descent for adjusting weights. The model was trained using different learning rates and epochs. They found out that to get high accuracy, a low learning rate and a high number of epochs should be used to train the model.

In the paper [10], the authors have discussed and compared traditional and deep learning-based approaches for anomaly detection. Traditional approaches use manual extraction of features and can't be generalized to work with scenes having unknown anomalies whereas deep learning-based approaches learn and extract features automatically. They can be generalized across multiple datasets. Finally, they talk about research issues like real-time processing, multi-view anomaly detection, etc. that need to be addressed.

In the paper [11], the authors have utilised a Deep Neural Network (DNN). Specifically, they have used a stack of CNN and ConvLSTM layers. The anomalies in the video are detected with the help of a regularity score which is the error between the predicted video sequence and the actual video sequence. Since the regularity score can be noisy, they have used the Persistence1D algorithm to detect only the useful minima. Their model could not outperform state of the art methods. But the proposed method has a few advantages. The feature extraction process is completely automatic; it does not contain any hand-crafted features. Another point is that the method does not have any dataset-specific parameters which make the method generalize well.

In the paper [12], the authors have proposed a new system. The input video is converted into image frames. These frames are then converted from RGB to grayscale. Then using the SIFT and Gabor method, features are extracted from the grayscale image. To classify the image, they use the SVM method. Their method separates successful data from video to anticipate movement in the ATM preface.

In the paper [13] ,authors used Tube Extraction module to get a spatiotemporal volume as output. The dataset is tested on full frame videos and action tubes. They got better accuracy on action tubes than full frame videos .Hence it was concluded that adding contextual information helps in anomaly detection.

The paper [14] gives details of different frameworks for object detection which can handle sub-problems such as occlusion, clutter and low resolution, with different degrees of modification on R-CNN.

In the paper [15], the model categorizes deep learning methods used for the detection of video anomalies into different groups and provides a comparative analysis for better selection of the methods that perfectly works for an application.

In the paper [16] provides a framework that uses binary features to encode temporal gradients with foreground features to classify events by relying on low complexity models. UCSD and LV dataset was used and both showed that framework outperforms online methods.

In the paper [17] authors used group activity recognition, Active region localization and fight action recognition to determine anomalies. They used FADS datasets, movies dataset, hockey dataset and UCF101 dataset.

In the paper [18] used deep multiple instance ranking framework with weakly labelled data to avoid labour - intensive temporal annotations of anomalous segments in training video. It is found that this is better than baseline methods.

In the paper [19], authors used CNN to extract information and max pooling is used to reduce computation load. SMTP is used to send video to host.

In the paper [20] uses five methods namely Deep Learning, Gaussian, Support Vector Machine (SVM), Fuzzy Logic, and Nearest Neighbour to detect the anomalies.

III. CHALLENGES FACED

The major problem we faced was finding the right dataset. We needed normal and abnormal videos from ATM CCTV cameras. But we couldn't find an appropriate dataset. At the best, we could find 16 to 20 abnormal videos from online websites. Since they have low resolution and very few videos, they cannot be classified as a good dataset. Also, creating a new dataset is a very time-consuming and laborious task.

IV. DATASET

The dataset we selected was the "Dr Chen anomaly surveillance dataset." We selected this dataset because the anomalies here were close to the anomalies which can be found in the ATM CCTV videos. We added the videos which we found from YouTube to this dataset. There are 50 normal videos (1.01GB) and 200 abnormal videos (5.99GB). The

abnormal videos are divided into 4 categories namely assault, abuse, arrest and arson. The total size of the dataset is around 7 GB.

V. PROPOSED METHODOLOGY

The proposed system includes five steps:

A. Data Preparation:

The dataset is divided in the ratio of 80:20. Training data occupies 80% and testing data occupies 20% of the dataset. The model will be trained using the training data and it will be validated using testing data.

B. Background Registration and Subtraction:

Using the background subtraction technique, we get the foreground data. This is the main part of the raw data which will get registered. The video in the training dataset is divided into multiple individual frames and are stored in a directory. The frame rate used is five. These frames are resized to make all frames have uniform dimensions. Then these frames are converted to grayscale with the help of a python library. This is done because classification with grayscale images gives higher accuracy than with RGB images across the several types of classifiers. These are then stored in a data structure which is a NumPy array here.

C. Developing the Model:

RNN (Recurrent Neural Networks) is one of the most prominent algorithms which has resulted in amazing achievements in the deep learning domain in the last few years. It was one of the first algorithms that remembers its input with the help of an internal memory. Since it can remember past data, it was the preferred algorithm for sequential data like time series, etc. The main advantage of RNN over the feed-forward neural network is that in the latter, data moves only in one direction i.e., from input layers to hidden layers and finally to the output layers. Hence the latter is bad at predicting and RNN was created to overcome this problem. RNN uses backpropagation through time (BPPT) to decrease error and learn. But RNN suffers from two main issues: exploding gradient and vanishing gradient. Exploding gradient occurs due to accumulation of large error gradients. This results in very large updates to the model weights during the training phase. Hence the model becomes unstable and is unable to learn from training data. Vanishing gradient occurs when small gradients accumulate due to the multiplication of small derivatives as we go down the layers. Due to this, the weights of the initial layers are not updated effectively and hence this increases the overall inaccuracy of the network.

To solve the problems of RNN, long short-term memory abbreviated as LSTM was created. It is basically an extension of RNN which enables RNN to remember inputs over long periods of time. This is because LSTM can save information in its memory. It assigns weights to data which helps RNN to either forget data, read new data or assign importance to data. This is done with the help of three gates namely input gate, forget gate and output gate. It solves the problems of RNN by keeping the gradient steep enough which keeps the training relatively short and accuracy high.

Spatiotemporal data contain feature information, such as temporal and spatial information, together. Therefore, spatiotemporal correlation patterns are often utilized together in prediction models. Since the RNNs do not consider the spatial structure, the spatial information within the data may be dropped during the pre-processing. The convolutional LSTM network recognizes the spatiotemporal correlation by combining the LSTM layer and the convolutional layer. This method can manage an individual entity and a group of entities that has a limited size. Therefore, we create a spatiotemporal auto-encoder architecture that is based on a 3D convolution network. The spatial and temporal information is extracted by the encoder, and reconstruction of the frames is handled by the decoder. The training is done using the previously created NumPy array.

The number of layers decides the efficiency of a deep neural network .We have used a 7-layer sequential model which consists of a linear stack of layers. The first or the initial layers of a CNN generally consist of convolution layers. These layers filter the images with the help of convolution kernels and return “feature maps”. An activation function is used to normalize the maps. The maps can also be resized. The first two layers are Convolutional 3D layers. The input layer if combined with the convolution kernel created by the above layers. The output is produced in the form of tensors input to produce a tensor of outputs. The first layer has 128 filters with a kernel size of (11, 11, 1). The second layer has 64 filters with a kernel size of (5, 5, 1). The activation function used is tanh. It is a non-linear activation function which is used in feedforward neural nets. Here, more positive input will be mapped near1.0 whereas more negative value will be mapped near -1.0 in the tanh graph. The next three layers are ConvLSTM2D layers. The first layer has a kernel size of (3, 3) with 64 filters and a dropout rate of 0.4. The second layer has a kernel size of (3, 3) with 32 filters and a dropout rate of 0.3. The third layer has a kernel size of (3, 3) with 64 filters and a dropout rate of 0.5. Here the input and recurrent transformations are both convolutional. We have not used separate Conv2D and LSTM layers. This is because Conv2D effectively captures spatial features, while LSTM is used to relations in data over time. Spatiotemporal features may not be detected properly if these layers are stacked together. The final three layers are Conv3DTranspose layers. This is required for upsampling without losing connectivity patterns. It does the opposite of normal convolution. The batch size used is one and the number of epochs is fifteen. The model is compiled using optimizer = Adam, loss = Mean Squared Loss and metrics = Accuracy. The Adam optimizer adaptively tunes the learning rate for each parameter during the optimization process using cumulative second-order statistics. Also, we have used early stopping for optimizing the training

of the model. It will check at end of every epoch whether the loss is decreasing or not. Once the loss is found to be no

longer decreasing, the training terminates. Mean squared error (MSE) is the most used loss function for regression. The loss is the mean overseen data of the squared differences between true and predicted values or writing it as a formula.

$$\frac{1}{N}\sum_{i=1}^{N}(y-\hat{y})^2$$

where ŷ is the predicted value.

D. Feature Selection:

An auto-encoder architecture based on a convolution 3D network is good in automatic feature extraction. The best part about CNN is that it learns the features automatically. The CNN itself carries out all the mundane work of extracting and describing features during the training phase. For optimizing the parameters of the classifier and the features, the error from classification is minimized. The convolution layers of the CNN perform this task. The purpose of this layer is to detect the features in the input images. It uses a technique called convolution filtering [Fig 2]. Convolution filtering uses the sliding window technique. The window represents a feature and this window slides over the image. Now we calculate the product between each portion of the image and the feature. This product is called as the convolution product. The convolution layer calculates the convolution of each of the input images with every filter. For each pair of images and filters, we get a feature map that tells us the location of the feature in the image.

E. Pattern Matching:

The abnormal events are identified by computing the reconstruction errors between the original and reconstructed batch. For this we have used mean squared error. It is calculated as the mean of the squared differences between the predicted and actual values.

To identify whether the given video is normal or abnormal, we determine it on the basis of the loss value. The loss is the mean overseen data of the squared differences between true and predicted values. The loss value is generated for every video given to the model. On the basis of the activities done in the video, the model generates a loss value. The system has a certain threshold for the loss value. If the given input video surpasses that threshold value, then it is considered as an abnormal video.

When we test the model with a video, if the model detects abnormal/anomalous activities, the video is sent to the dashboard. We can view all the abnormal videos on the dashboard. Also, an alert is sent to the email address registered. The alert will contain a message stating that an abnormal activity was detected.

Here is the abstracted flow of the process for clear understanding [Fig 1]:

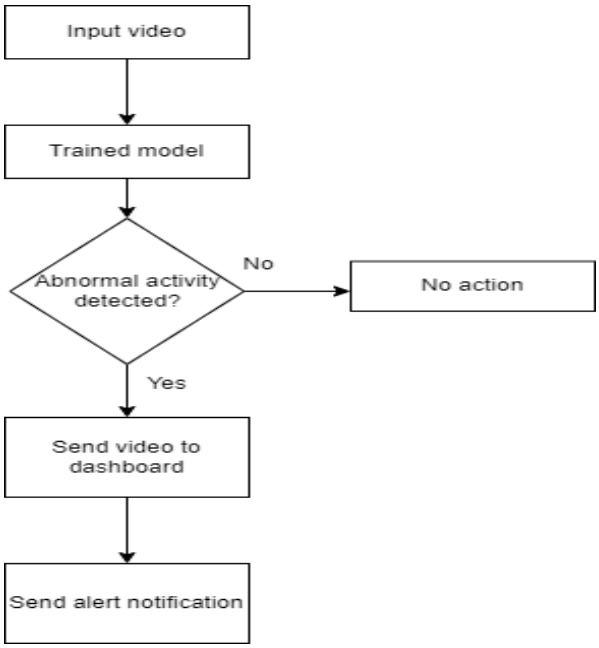

Fig 1. The flow of the process the Flowchart depicts the abstracted flow of the system when a video is sent as an input.

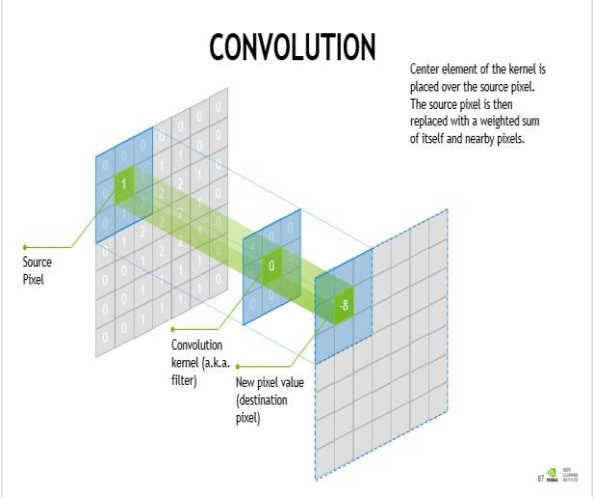

Fig 2. Convolution filtering. It depicts the process of mapping the source pixel to the new pixel value with the help of a convolution kernel [21].

VI. EXPERIMENTAL RESULTS

We trained the model using the training dataset using 15 epochs. Next, we test our model using the testing data which we generated by dividing the dataset in the ratio 80:20.

Test on abnormal video:

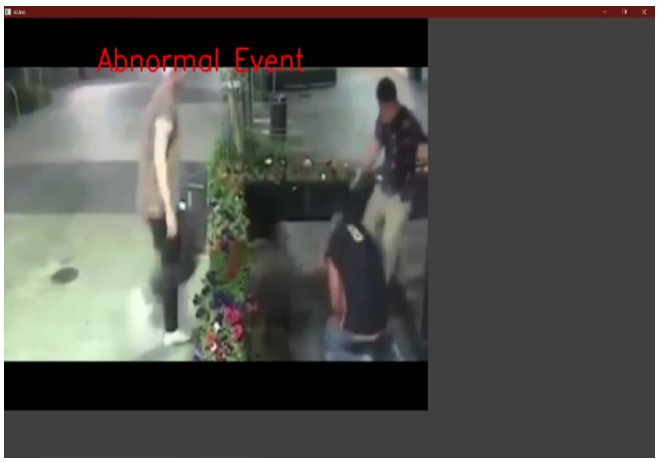

Fig 3. Video popup showing abnormal event detected. Since the model detects an abnormal event in the input video, we get a message written on the popup stating that an abnormal event was detected.

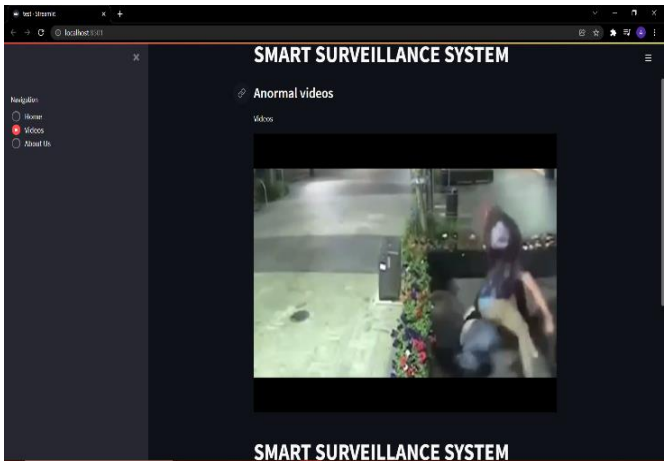

Fig 4. Video sent to the dashboard. Since abnormal event was detected in the input video, it is sent to the dashboard and is displayed.

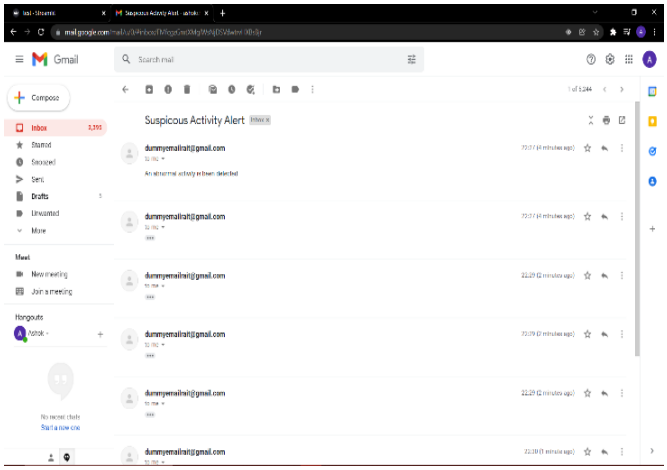

Fig 5. Alert sent to the registered email address. We get this alert because an abnormal event was detected in the input video.

Test on normal video

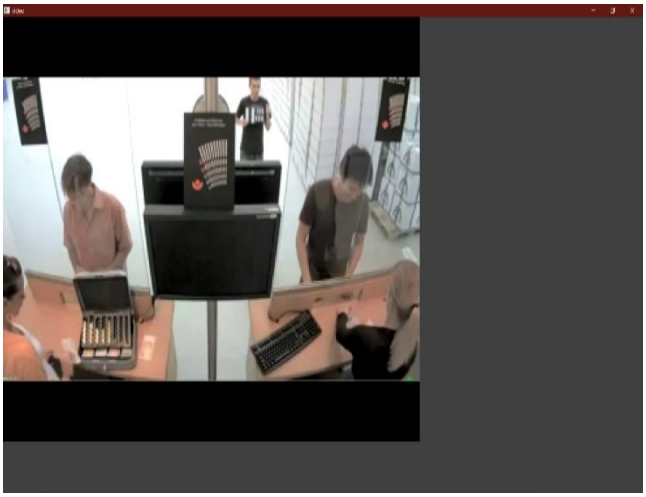

Fig 6. Video popup. Since no abnormal event was detected, there is no message displayed on the popup.

When we test our model on the abnormal video, we get a video popup showing that abnormal activity is detected. Simultaneously the video is sent to the dashboard and an alert notification is sent to the registered email address stating that an abnormal activity is detected. Similarly, when we test a normal video, we also get a video popup. Since the video has no abnormal events, we do not send the video to the dashboard and no alert is sent.

TABLE I
ANALYSIS OF 5 EPOCHS

	Correct	**Incorrect**	**Total**
Abuse	2	7	9
Arrest	7	2	9
Arson	4	4	8
Assault	7	7	14
Normal	10	2	12
Total	30	22	52

Table I shows the analysis of 5 epochs on our training model. The model is been analysed on 5 parameters. Total 52 videos were used here. The model correctly classifies 30 videos on the given parameters while it gives incorrect output for 22 videos. So from this we get an accuracy of 57.7% for 5 epochs.

TABLE II
ANALYSIS OF 15 EPOCHS

	Correct	**Incorrect**	**Total**
Abuse	5	4	9
Arrest	7	2	9
Arson	5	3	8
Assault	7	7	14
Normal	11	1	12
Total	35	17	52

Table II shows the analysis of 15 epochs on our training model. The model is been analysed on 5 parameters. Total 52 videos were used here. The model correctly classifies 35 videos on the given parameters while it gives incorrect output for 17 videos. So, from this we can accuracy of 67.30% for 15 epochs.

VII Conclusion

In this paper, we have attempted an approach for identifying the anomalies in the given video like theft, assault, robbery etc and to immediately notify the respective authority through an email. In the first step we divided the data into training and testing in the ratio 80:20, then by using background subtraction we got the foreground data. The model was developed and by using auto-encoder architecture feature extraction was done. Finally, the pattern matching is done where abnormal events are identified. We have used the dataset of Dr Chen which is of 7 GB. The proposed technique gave an accuracy of 57.7% for 5 epochs and 67.30% for 15 epochs.

VIII Specifications

Python libraries used – TensorFlow, Keras, cv2, streamlit

GPU – Nvidia RTX 3050 4GB

CUDA – 11.2

Cudnn – 8101

Ffmpeg 2021

IX References

[1] Wu, Guangli, et al. "Video Abnormal Event Detection Based on CNN and LSTM." *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2020.

[2] Sreenu, G., and MA Saleem Durai. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis." *Journal of Big Data* 6.1 (2019): 1-27.

[3] Nasaruddin, Nasaruddin, et al. "Deep anomaly detection through visual attention in surveillance videos." *Journal of Big Data* 7.1 (2020): 1-17.

[4] Anil, Rohan, et al. "Memory efficient adaptive optimization." *Advances in Neural Information Processing Systems* 32 (2019).

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

[6] Kakadiya, Rutvik, et al. "Ai based automatic robbery/theft detection using smart surveillance in banks." *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2019.

[7] Yang, Biao, et al. "Anomaly detection in moving crowds through spatiotemporal autoencoding and additional attention." *Advances in Multimedia* 2018 (2018).

[8] Pathak, Ajeet Ram, et al. "Assessment of object detection using deep convolutional neural networks." *Intelligent Computing and Information and Communication*. Springer, Singapore, 2018. 457-466.

[9] Janakiramaiah, B., G. Kalyani, and A. Jayalakshmi. "Automatic alert generation in a surveillance systems for smart city environment using deep learning algorithm." *Evolutionary Intelligence* 14.2 (2021): 635-642.

[10] Pawar, Karishma, and Vahida Attar. "Deep learning approaches for video-based anomalous activity detection." *World Wide Web* 22.2 (2019): 571-601.

[11] Singh, Prakhar, and Vinod Pankajakshan. "A deep learning based technique for anomaly detection in surveillance videos." *2018 Twenty Fourth National Conference on Communications (NCC)*. IEEE, 2018.

[12] Arpitha K, Honnaraju B "Vision-Based Anomaly Detection System for ATM" International Research Journal of Engineering and Technology (IRJET), 2018.

[13] Landi, Federico, Cees GM Snoek, and Rita Cucchiara. "Anomaly locality in video surveillance." *arXiv preprint arXiv:1901.10364* (2019).

[14] Zhao, Zhong-Qiu, et al. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30.11 (2019): 3212-3232.

[15] Nayak, Rashmiranjan, Umesh Chandra Pati, and Santos Kumar Das. "A comprehensive review on deep learning-based methods for video anomaly detection." *Image and Vision Computing* 106 (2021): 104078.

[16] Leyva, Roberto, Victor Sanchez, and Chang-Tsun Li. "Fast detection of abnormal events in videos with binary features." *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[17] Xu, Qichao, John See, and Weiyao Lin. "Localization guided fight action detection in surveillance videos." *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019.

[18] Sultani, Waqas, Chen Chen, and Mubarak Shah. "Real-world anomaly detection in surveillance videos." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

[19] Malekar, Mrunal. "Detecting Criminal Activities of Surveillance Videos using Deep Learning." (2021).

[20] Shidik, Guruh Fajar, et al. "A systematic review of intelligence video surveillance: Trends, techniques, frameworks, and datasets." *IEEE Access* 7 (2019): 170457-170473.

[21] https://blogs.nvidia.com/blog/2018/09/05/whats-the-difference-between-a-cnn-and-an-rnn/.