Deep Neural Network based Anomaly Detection for Real Time Video Surveillance

D A Sneha
sneha.da2020@vitstudent.ac.in
Integrated Software Engineering, Vellore Institute of Technology, Chennai, India

John Raj I
john.raji2020@vitstudent.ac.in
Integrated Software Engineering, Vellore Institute of Technology, Chennai, India

Lakshya
vlasayasvj1307@gmail.com
Integrated Software Engineering, Vellore Institute of Technology, Chennai, India

Abstract

One of the main concerns across any domain has been the need for security. With the crime rates increasing every year the need to control this has become critical. Video surveillance is one among the various methods available to monitor crime or any anomalous behavior. Nowadays security cameras capture incidents in almost all public and private places if desired. Even though we have an abundance of data in the form of videos they need to be analyzed manually. This results in long hours of manual labor and even small human discrepancies may have huge consequences negatively. The focus of this work is to build a model with convolution neural networks to detect any form of abnormal activities or anomalies in the video. The model will convert the input video into frames and will detect the anomalous frames. To increase the efficiency of the model, the data is de-noised with Gaussian blur function. As part of this work two datasets namely avenue dataset and anomaly detection dataset were used to predict various kinds of anomalies. The performance of the model is measured through accuracy of classification and the results are presented.

Keyword: Deep learning, Convolution Neural network, anomaly detection, video surveillance, Deep Neural network

Introduction

Computer vision is one of the most rapidly developing technologies that collects information from all forms of images and videos. Over the past decade or so, research in this area has increased rapidly, and it has been broadly deployed in navigation, self-driving cars (i.e., visualization), surveillance and many similar tasks. Computer vision is a subset of artificial intelligence and it helps us to understand and evaluate the real world. And by using deep learning models with computer vision, our systems can recognize objects from digital images and respond accordingly to the images recognized by our system. And the researchers working on simulating human visual systems can actually automate all works in the future which need visual cognition. But when a huge amount of data is used, then interpreting the image becomes too complex when compared to a separate form of binary information. This clearly depicts that in a few years computer vision would be capable enough of emulating human vision and identifying its own patterns from images, which could be compared with the human visual cognitive system.

To understand the sense of the pictures in an exceedingly far better way, a deep learning algorithm called convolutional neural networks (CNN) is designed. Further, these neural networks are trained by giving in an exceedingly huge number of images that will help to amass and determine everything that's present within the images with the utilization of deep learning algorithms. After this, the network examines the pictures' pixel by pixel and determines the expected output with the knowledge found with the massive number of images given as training before as within the case of supervised learning.

Nowadays big data applications are popular across the industry and research areas. Among these widespread samples of big data, the role of video streams using CCTV cameras is equally important because of the other sources like social media data, sensor data, agricultural data, and so on. CCTV has been implemented in areas where security is crucial and moreover manual surveillance may be a very tedious work which always consumes lots of your time. Security is often defined in many alternative terms as in terms of violence, robbery, accidents, and then on. But when a crowded place is taken into account, security covers most kinds of abnormal events. This work includes deep-rooted research on how anomalies in crowd areas will be found and also the majority of the works reviewed during this survey are supported deep learning techniques. Various deep learning methods are compared in terms of their algorithms and models.

Many recent acts like the terrorist attacks and lots of other factors have highlighted the urgent need for efficient surveillance and thus security and surveillance are the foremost important issues in today's world and thus, we wanted to try and do our research in this particular field. Therefore, the traditionally used contemporary surveillance systems were used for surveillance which uses digital video (DVR) cameras that play host to multiple channels. But the most important drawback of this model is that it requires continuous human monitoring which causes human fatigue and therefore the cost of labor. To understand the time during which the anomaly went on, we'll manually undergo the total video footage which is tiring, hence we all need an automatic system that may detect anomalies on its own.

In our work, we have performed image enhancement for every frame to make it easier for the model to process the data. We have improved the accuracy of our model by applying a few enhancement techniques such as gray scale and gaussian blur. This resulted with an accuracy of 74.5602%. It is to be noted that the accuracy before applying the enhancements was 49.2102%. This increase in accuracy depicts the significance of data preprocessing techniques. We have used Gaussian Distribution to plot the mean of the normal video.The dataset used for the work is Anomaly Detection Dataset collected by Dr.ChenChen. The reported accuracy after applying C3D feature extraction on the dataset is 23.6%. However our method outperforms the existing method with an accuracy of 54.5602%.

Literature Review

Jianting Guo. et.al, [1] have focused on the parameter estimation and anomaly detection in an encrypted video bitstream. The anomalies are detected at the frame level using feature extraction techniques. Adaptive kernel density estimator is used to estimate the probability density function of the extracted feature. They have used four public datasets namely Avenue, Subway, UCSD, and UMN datasets. Though various models were used,they have obtained accuracy of 80%(Avenue) and 90%(UMN) in scheme [29] and 77%(Avenue) and 88%(UMN) in scheme [42].To make their model more economical they have excluded video decryption, full decompression or any interactive protocol.Joey Tianyi Zhou. et.al, [2] have proposed a model with neural networks for anomaly detection by deeply achieving sparse representation, feature learning, and dictionary learning in the three joint neural processing blocks. The feature transfer block is employed to extract discriminative features by exploiting the interchangeable ness of the neural network from totally different tasks/domains. They have used three datasets namely the UMN datasets, UCSD Pedestrian, and CUHK avenue. Though various models were used,they have obtained highest accuracy of 87.5% using Resnet-152 model.They increased the efficiency by capturing motion, eliminating noisy background, and alleviating data deficiency specifically by designing a motion fusion block accompanied by a feature transfer block. They are yet to design a novel recurrent neural network to embrace the merits of neural networks and learn sparse coding optimizers.

Weixin Luo.et.al, [3] have presented a TSC framework for anomaly detection. This preserves the similarities between frames. Their model with TSC can be interpreted with special sRNN. By optimizing all parameters in sRNN-AE at the same time, they avoided the nontrivial parameter choice and reduced the procedure value for inferring the reconstruction coefficients within the test phase. They have used five datasets namely the UCSD Ped1, UCSD Ped2, CUHK avenue, Subway Enter, Subway Exit dataset. Though various models were used,they have obtained highest accuracy of 83.48%(Avenue), 92.21%(Ped2), 85.38%(Entrance), 89.73%(Exit) and 69.63%(their dataset) using sRNN-AE model.The merits of this work are that they have built a new dataset which is the most challenging one in terms of data volume and scene diversity.Kai-Wen Cheng. et.al, [4] proposed an approach to detect global and local anomalies have proposed a hierarchical framework via stratified feature illustration and Gaussian method regression (GPR). This method is statistical and supports sparse features. It does not fail due to noisy training data. They have evaluated the proposed method based on four widespread real-world datasets namely the Subway, Behave, UCSD Ped1, and QMUL Junction datasets. Their technique is able to achieve 85% detection rate by training with 4 difficult datasets.

Wenqing Chu. et.al, [5] have proposed a method that is based on both, optimizing the sparse coding and unsupervised feature learning for video data. In this, three datasets: Avenue, subway, and UCSD are used. Our approach learns effective discriminative features.Though various models were used,their model has obtained highest accuracy of 82.1%(Frame-level) and 93.7%(Pixel-level). However, the performance is not so good. Whereas, supervised deep neural networks have achieved great progress. One disadvantage is that their framework has not been extended to other video

analysis tasks that involve expensive labelling effort.Yuanyuan Li.et.al, [6] in their work adopt a structure to restructure a current input frame or predict a future frame. An auto-encoder (which is the adopted structure) consists of an encoder and a decoder. The study work is a novel spatiotemporal U-Net.Though various models were used,their proposed model obtained accuracy of 83.8%(Ped-1), 96.5%(Ped-2) and 84.5%(Avenue). In this method, it is possible to predict frames using normal events and detect abnormality using prediction error. It includes the advantages of U-Nets in representing spatial data with the capabilities of ConvLSTM for modeling temporal motion knowledge.

Roberto Leyva. et.al, [7] have proposed a work 'Video Anomaly Detection with Compact Feature Sets for Online Performance'. Their features are extracted from a unique cell structure that helps to outline support regions in a coarse-to-fine fashion. To increase detection accuracy, they consider the joint response of the models in local spatio-temporal neighborhood. They test this on various datasets so that their methodology can be implemented to find anomalies in real life surveillance using CCTVs.The datasets used by them are UMN dataset and the UCSD dataset. They have used gaussian mixture models and the highlight of their work is that a variable sized cell structure allows frame extraction from a limited number of support regions thus reducing the amount of frames that need to be processed. Their significance is that they have bought a accuracy of 88.3% with a frame processing time of just 30ms.

Thittaporn Ganokratanaa. et.al, [8] in their work 'Unsupervised Anomaly Detection and Localization Based on Deep Spatio Temporal Translation Network' have proposed Generative Adversarial Network (GAN), a Deep Spatiotemporal Translation Network (DSTN), localization method based on Edge Wrapping (EW) and novel unsupervised anomaly detection. They generate dense optical flow as temporal features using normal frames alone. The model is trained with traditional normal videos and during testing, the video sequences are given into the model as input and abnormal videos are classified if they vary from the trained nirmal pattern. They have used the datasets UCSD pedestrian, UMN, CUHK Avenue datasets and the highlight of their work is that by incorporating domain transformations, this proposed method expands the capabilities of low rank sparse representation methods, such as mcRoSuRe-A. These methodologies help us to find more precise correspondence between different parts of the data matrix. The accuracies for the ped1 dataset, ped2 dataset and the avenue datasets are 83.46%, 93.06% and 84.25% respectively.

Siqi Wang. et.al, [9] has proposed a system in which unsupervised settings are made with very few attempts that detect irregularity without knowing common events in advance. They used Avenue, UCSD ped1, UCSD ped2, Subway Exit datasets. And they used Deep Spatio Temporal Translation Network (DSTN). Unsupervised strategies which actively notice irregularity forcefully for local changes that fail to notice the worldwide topological context. A new unsupervised perspective is proposed, and this perspective keeps away from physically specifying normality for instructing as supervised methods do and also takes the whole Spatio-temporal context into consideration. The proposed DSTN framework is embedded with concepts of deep convolutional neural network of GAN based Edge Wrapping approach which brings advantages to anomaly localization.

Eric Jardim.et.al, [10] proposed a system for Anomaly Detection in Moving-Camera Videos using Domain-Transformable Sparse Representation. Their work uses sparse representation that detects anomalies in video sequences generated with moving cameras to present a special matrix factorization. By associating the frames of the target video above representation is made, that is the frames of an anomaly-free reference video, which is a previously validated sequence with a sequence to be tested for the presence of anomalies. They used VDAO-200 dataset with an autoencoder system in which they used a two-stage approach to detect anomalies. First we use unsupervised learning to train the model to predict normal videos and later on acquire a refined model to detect the abnormalities.Nanjun Li.et.al [11] have proposed a Spatio Temporal Cascade Autoencoder. This makes use of a cuboid patch-based method which successively makes full use of spatial and temporal cues from video data. Three datasets, namely the UCSD, Avenue, and therefore the UMN dataset are trained using these models. This model is widely accustomed to detect and locate multiple sorts of abnormal behaviors in real-world videos. And then they had a higher of higher accuracy of 87.1% using the methodology ST-CaAE.

Rashmika Nawaratne.et.al [12] wanted to handle the constraints and challenges in anomaly detection and localization in real-time video surveillance. This can be dispensed on the UCSD (Ped 1 and Ped 2) and CUHK datasets. The first challenges faced during this are handling high-dimensional video in real-

time, model learning normality, and adapting with the assistance of fuzzy aggregation and active learning to normal behavior that continuously evolves. The ISTL approach proposed in the paper is based on the spatiotemporal autoencoder model, consisting of convolution layers. This model preserves the spatial structure of the video stream. The accuracies for the ped1 dataset, ped2 dataset and the avenue datasets are 81.00%, 91.10% and 80.6% respectively.

Tao Xiang. et.al [13] in their paper have addressed the issues of modeling video behavior. this can be later accustomed detect normal behavior and anomalies. They have developed a unique framework for this purpose that does not require labelling of the dataset. The key components of the framework are Dynamic Baysian Network(DBN), grouping and generalising normal behaviour patterns and finally detecting normal behaviour method using LRT method. For this experiment the authors have collected and used data from a fixed CCTV camera, mounted at a craft moorage space which observes the craft moorage procedure.Through various experiments, it was seen that a behavior model trained using unlabeled data gave a higher anomaly detection rate(79.2%) and a much lesser false alarm rate(5.1%) compared to the model on labelled data(anomaly-71%, false alarm - 12.4%) and hence outperforms the latter in detecting abnormal behavior patterns.

Lucas A. Thomaz. et.al [14] have proposed a family of algorithms that are based upon sparse decomposition. These algorithms detect anomalies in slow-moving video cameras. The primary step within the algorithm is to compute the union of subspaces that best represents all the frames from a specific reference video. Then a low-rank representation of the target video is performed. The database used here is the VDAO database. One major drawback is that the algorithms have high computational complexity and hence higher cost. The authors have tried to combat this problem by using intrinsic properties in the data and restricting the search space. Through experiments the authors were able to show that the mcRoSuRe-A method was able to run about 2.6 - 100 times faster than the other methods depending on the size of the video.Weixin Li et. al [15] in their paper discuss a joint detector of spatial and temporal anomalies. The dataset used here is UCSD, UMN, and others. Here crowded scenes are sculptured with a hierarchy of MDT models, temporal anomalies are equated with background subtraction. Similarly, spatial abnormalities are equated to discriminant salience, and joining the abnormal scores across space, time, and scale with a CRF. A joint representation of video appearance and dynamics and globally consistent inference are used so that the anomaly detector can span through time, space and spatial scale. The detector was able to achieve 99.5% accuracy.

Ke Xu, Xinghao Jiang. et. al [16] have explained Stacked Sparse Coding (SSC) with an intra-frame classification strategy. The datasets used are the Avenue dataset, UCSD Ped2 dataset, UCSD Ped1 dataset, and Subway dataset. Here, all spatial and temporal features within the sub-regions are described using a FIP descriptor. This is done so that detection and localization results can have higher accuracy. SSC encoding will encode both spatial and temporal connections of subregions. Finally, the intra-frame classification strategy improves detection performance. Limitations in this method include frame by frame detection which makes it discontinuous.Yiru Zhao. et.al [17] have proposed Spatio-Temporal Autoencoder. This model is constructed using deep neural networks to find out and interpret videos automatically. Feature extraction from both spatial and temporal dimensions is finished using 3D convolutions. Here the UCSD dataset CHUK dataset and therefore the Traffic dataset, collected by the authors, is used. The proposed method (STAE-3d-decreasing-pred) gives 11.3% increase in AUC and 23.4% decrease in ERR when compared to ConvAE.

Yuan Yuan. et.al [18] have proposed a paper that mainly focuses on the abnormal detection of crowd behavior. UMN and UCSD datasets were used. One of the highlights of the paper is that, this is the first time visual structural context of videos was explored in this field. Crowd abnormality detection was done using online spatial-temporal analysis of the SCD variation and it outperformed other methods.The model however was tested only within the visible video sequences containing RGB channels. In extreme and bad weather conditions, the model may falter.Ata-ur-Rehman. et.al., [19] have proposed a method that detects abnormal video frames based on posterior probability (determined with size, motion and location features) throughout a video sequence. It detects all the video frames with abnormal activities. It also detects abnormal regions within abnormal video frames. This work was carried out using UCSD and LIVE datasets. The proposed model outperforms existing models in terms of AUC and ERR and gives very low processing time.

Proposed System

As we all knew it is very tiring to manually monitor all the anomalies in crowded places, thus we need to build a system that would find exactly when the anomaly happens on its own but with a higher accuracy and better model creation and better pre-processing when compared to the previous works done in the same concept. The block diagram clearly depicts that the pre-processing of both the training and test sets are done in the same way. Both undergo a FFMPEG video extraction which converts them into frames and into gray image. Then denoising is done using gaussian blur and then the training set is fed into a model with three sub models (Conv3D, ConvLSTM2D, Conv3DTranspose) and with that we train our model and then then the test set is given into the model and a value for loss is obtained which corresponds to the behavior of the video. If the loss is greater than the threshold then it is called the anomaly video, else it is considered to be a normal video.

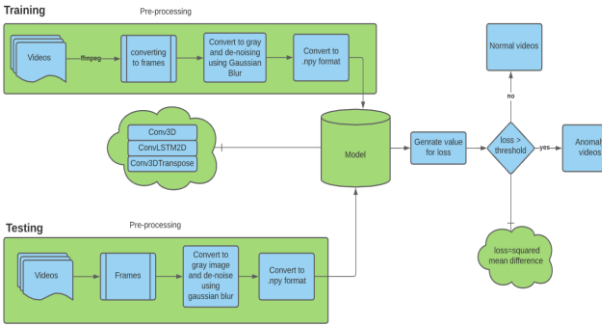

Figure 1: Block Diagram of Proposed Anomaly Detection System

In this work, Anomaly-detection-dataset and Avenue dataset are used. Here only 80% of the normal training videos is used to train the model. As the videos are pre-processed in such a way that the videos are split into frames by using FFMPEG that may well be a command-line tool that converts audio or video formats for video frame extraction, then each frame gets converted into a gray frame and each frame is gone through a denoising process for which we have used Gaussian Blur. Then these frames are converted into .npy format and they are stored together in a dataframe named imagestore which was initialized with an empty array. This dataframe is fed as the input to our model and in our proposed model, we have given all normal videos alone and from the given imagestore (dataframe), we use CNN to predict a value of loss. The loss is the mean squared difference between the actual normal frame and the normal frame obtained in the hypothesis. And we get different values of loss for each frame. And we will have to manually choose a loss value above which if any video has a loss, then those videos will be called the anomaly videos and the manually chosen loss value is called the threshold.

It is the same as the traditional Gaussian analysis where we draw a graph for the given normal videos and if our tested value is much deviated from the mean value then it is called the anomaly video. It's the same here as well. And then the remaining 20% of the normal training videos are taken as the validation set and the same preprocessing which we did for the training set is done here and then they are converted into a .npy format and it is given into the same model and the loss is calculated and if the all losses are lesser than the threshold, then our model is working here. If not, try to increase accuracy by fine tuning the hypo parameters and changing the value of the threshold. And then our model is ready to test. All the other videos in the dataset can be given as the testing videos and the same pre-processing and conversion to .npy format is done here and then it is given to same model to predict the value of loss and if loss>threshold for even any one frame of the mode, it is classified as the anomaly video.

The most important and crucial part of our work is to calculate the loss function using the squared mean error method, and generally squared mean error is calculated as in equation 1

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \underline{y_i}\right)^2$$

(1)

And here "y" is the actual expected output of the model and "y'" is the calculated output from the model as in equation 2.

$$\underline{y} = model.predict(y)$$

(2)

And here both the expected and the calculated outputs are in a .npy format. (i.e., y and y' are in numerical array format which consists of many values regarding to the image)

To explain our mechanism much easier I would consider a straight line which would fit all the y values in it (i.e., considering linear regression to

explain the derivation of the equation) as in equation 3.

$$MSE = \left[\left(y_1 - (mx_1 + b)\right)^2 + \left(y_2 - (mx_2 - b)\right)^2 + \cdots + \left(y_n - (mx_n - b)\right)^2\right]/n \quad (3)$$

The same mechanism is used here as well, but instead of using a straight line "y=mx+b" to find the training examples, we have used a CNN model named "LSTM" to predict the output.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} \quad (image.npy - model.predict(image.npy))^2 \quad (4)$$

And then this mean squared error fiction returns a value which is stored in the variable loss for further classification of the videos as in equation 4. The algorithms pertaining to training and testing are listed below.

Algorithm for Training

Step 1: Input the video by mounting the respective dataset from the drive.

Step 2: Import all the necessary libraries.

Step 3: Define the required functions such as create_dir(), remove_old_images(), store(), load_model() and mean_squared_loss().

Step 4: After defining the source path for the training videos, separate each video into frames using ffmpeg which is a command-line tool that converts video formats into frames.

Step 5: Convert each frame to a gray image and use Gaussian Blur to remove noise from the frame to increase the efficiency of our model.

Step 6: Display the subplots using the imshow() function from OpenCV.

Step 7: Convert them to .npy format

Step 8: We then split the data into training and validation splits.

Step 9: Give the training split videos as an input to our LSTM model to train the model with the most suitable hyper-parameters which

- Develops a model in which the loss is minimized and accuracy is increased.
- Loss is exactly the difference between the mean and hypothesis value in a Gaussian analysis.

Step 10: Next, the graphs are plotted between Training vs Validation accuracy and Training vs Validation loss.

Step 11: The model is set ready for testing.

Algorithm for Testing

Step 1: Train the model using the validation split.

Step 2: Separate each video into frames much similar to training.

Step 3: Convert each frame to a gray image and use Gaussian Blur to apply denoising to the frame.

Step 4: Convert them to .npy format.

Step 5: Give it as an input to the trained LSTM model, so that a value of the loss for each frame is fetched.

Step 6: Even if the value of loss of one frame is higher than the threshold then that video is labelled as an anomaly.

Step 7: Else the video is labelled a normal video.

Step 8: Then the subplots are plotted which is labelled whether the video is either anomaly or normal.

Step 9: Finally, a random video is tested and the output is respectively shown.

Experimental Setup

Google colaboratory was used to train the model. Also, FFmpeg software was used to convert the videos into frames. Pre-processing and de-noising the frames was carried out with the help of OpenCV library. In this research work, we have considered the part where our model determines a value for loss using the squared mean difference function as the threshold because the classification is completely dependent on the loss value. With all the loss values of our training examples, we should manually choose a value and that value will be named "threshold". And if the loss value of a video is less than the threshold value then it means that the difference between the expected answer and the obtained answer is less, therefore the video will be considered a normal video. Suppose the value of loss for any frame in the video is greater than the

threshold, the frame doesn't follow the normal frame pattern, it differs from the normal frame pattern a lot and thus it is considered to be an anomaly frame. If the video has even one anomaly frame. Even then the video is considered to be an anomaly video. And the main part here is choosing the threshold, the threshold can be chosen by observing the loss values of all the frames in the training videos. Because choosing a lesser value of threshold, then the result will become true negative by detecting the normal videos as anomaly videos. And if at all if we choose a higher value of threshold, again the result will become true negative by detecting the anomaly videos as normal videos. Thus, choosing the value of threshold is the key factor of determining the anomaly videos in our work. This work used two projects namely, Avenue dataset and Anomaly detection dataset.

Avenue Dataset -There are 16 training and 21 testing video clips. The model is trained with these videos and used over the test data to predict anomalies. The testing video clips contain 47 anomalous activities mostly by pedestrians which includes skating, throwing papers, people walking in the grass etc.

Anomaly detection dataset – This is a huge dataset that consists 1900 real world surveillance videos different anomalous activities. They are further grouped into 13 labels based on the crime committed. This is said to be one of the largest video datasets with a total of 128 hours of videos. Each video clip is mostly in the time range of 1-3 minutes.

Results and Discussion

As part of this research work, two different datasets namely Anomaly Detection Dataset and Avenue Dataset were used to detect anomalies. In order to fit the data to the model, a model.fit_generator function which is already built in the keras module was used. The "model.fit_generator" function can exploit the hyper-parameters by constantly changing them and observing the accuracy of changing them. A few readings taken while changing the hypo parameters as tabulated below in table 2.

Table 2: Comparative analysis

As shown below in the tabular column, we can simply observe the change in accuracy by changing the hyper parameters. We also tried making changes in the number of hidden layers in the model. Changing the number of hidden layers did affect our accuracy in a very minimal amount, thus changing the hidden layers wasn't that useful and we will try implementing other models in our future work to observe if there is a change in the accuracy significantly.

Figure 2: Sub-plots after pre-processing

From figure 2, we infer that the subplots of the original frames are converted to the Gray scaled and Gaussian blur and then displayed for the Anomaly Detection Dataset.

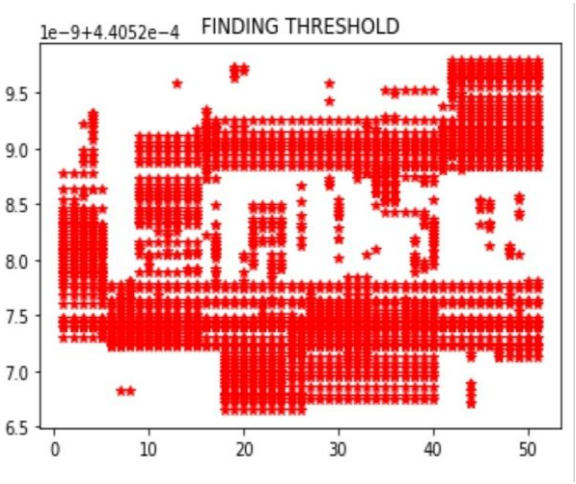

Figure 3: Finding threshold

From figure 3, we infer that the threshold is one of the critical aspects which is determined by plotting the above graph and the value for threshold for testing is initialized using the 1e-9+4.405e-4.

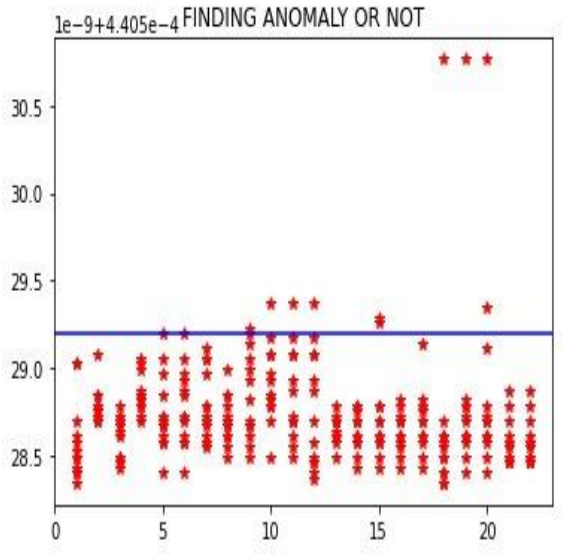

Figure 4: Finding Anomaly or not

From figure 4, we infer that all the points above the blue line (threshold) are detected as anomaly and others are detected s anomaly videos.

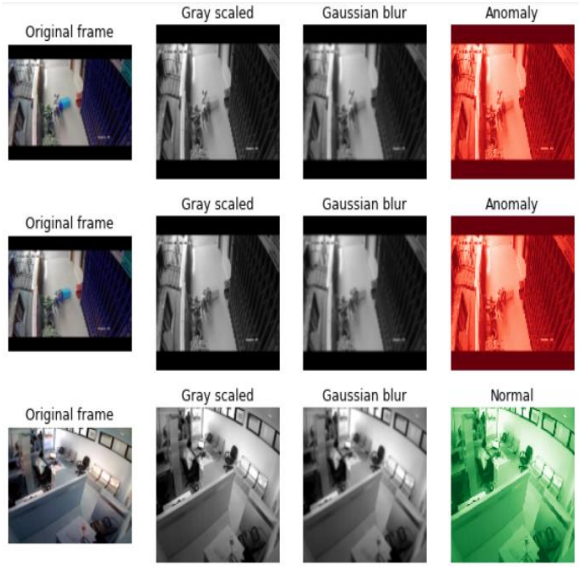

Figure 5: The subplots of resulted output

From figure 5, we infer that the subplots of original frames, Gray scaled, Gaussian blur. If the anomaly is detected, it is displayed in red color else it is detected as normal color is displayed in green color.

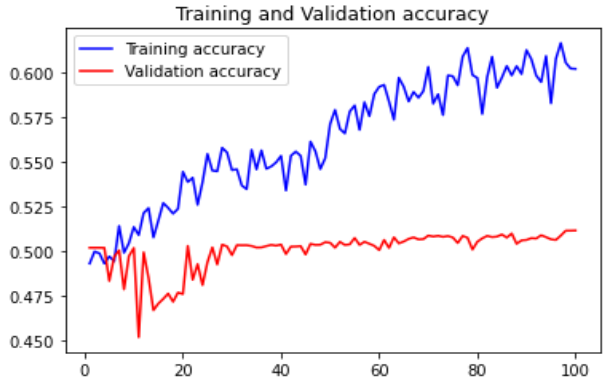

Figure 6: Graph plotted between Training accuracy vs Validation accuracy

From figure 6, we infer that the training accuracy better than validation accuracy, model can predict training data better and validation accuracy stagnates and is also noisy in the initial part of the graph.

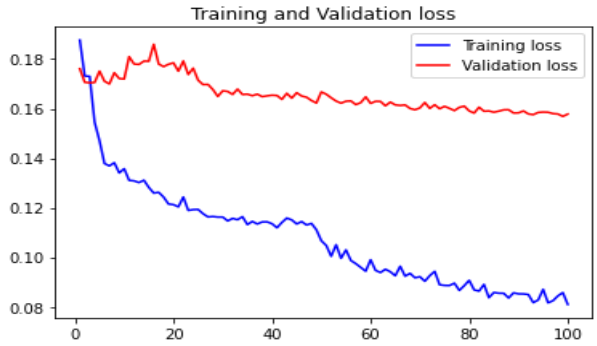

Figure 7: Graph plotted between Training loss vs Validation loss

From figure 7, we infer that the model can be given more training, training has halted prematurely and training set is easier to predict than validation set.

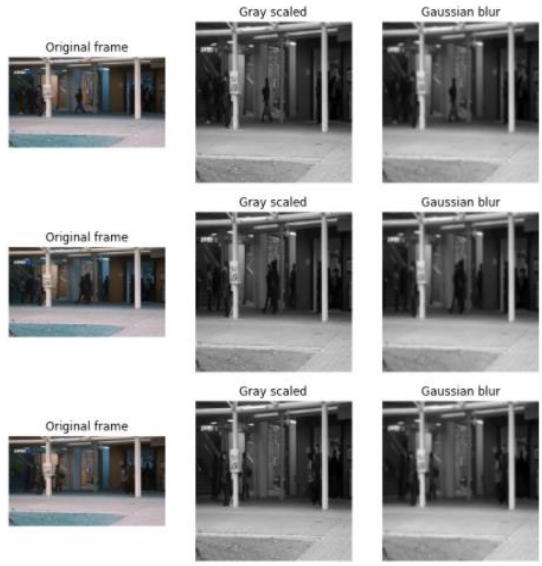

Figure 8: Sub-plots after pre-processing

From figure 8, we infer that the subplots of the original frames are converted to the Gray scaled and Gaussian blur and then displayed for the Avenue Dataset.

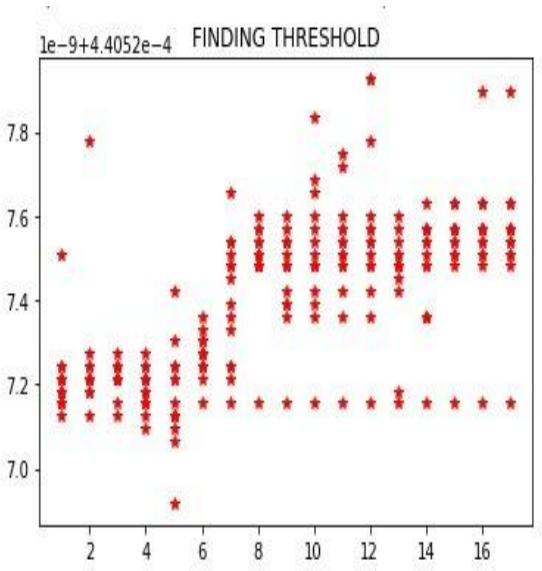

Figure 9: Finding threshold

From figure 9, we infer that the loss of our training examples is somewhere between 0.0004405292 and 0.0004405265. Thus, we choose the maximum value of the loss above which if a frame has a loss value, then the frame is an anomaly frame. Thus, we chose 0.00044052925 as our threshold value.

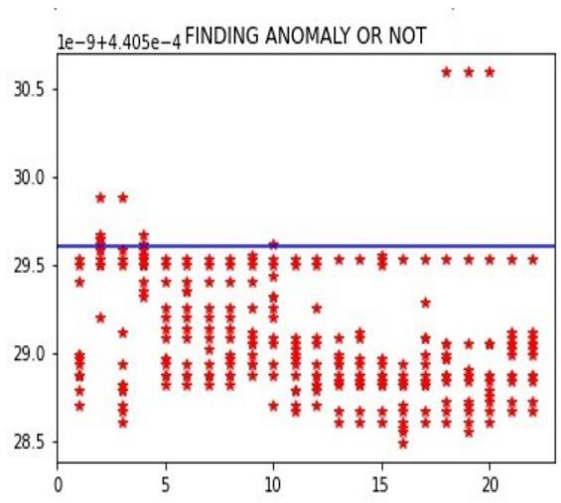

Figure 10: Finding anomaly or not

From figure 10, we infer that the blue line indicates the threshold and for frames having loss above threshold (i.e., above the blue line) is considered to be an anomaly frame. Even if one frame in a video is found anomaly, then the complete video is to be considered as an anomaly video.

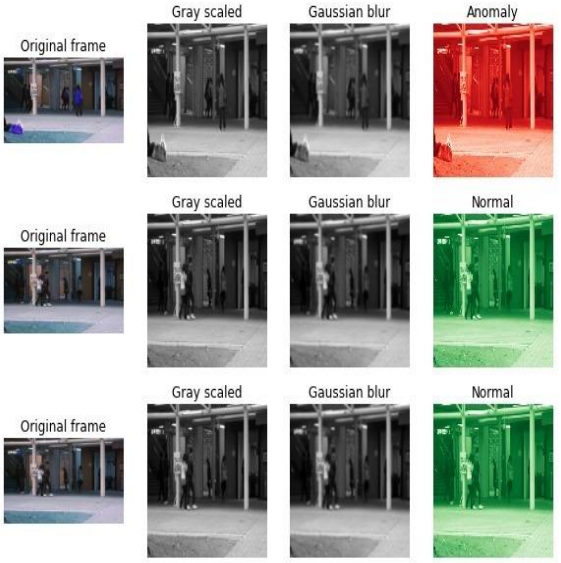

Figure 11: The subplots of resulted output

From figure 11, we infer that we have made a visualization of which frame is anomaly and which isn't.

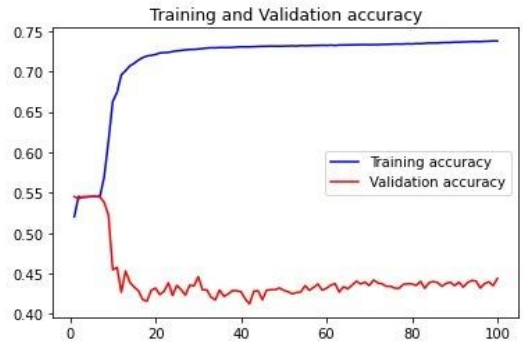

Figure 12: Graph plotted between Training accuracy vs Validation accuracy

From figure 12, we infer that these plots and graphs, we infer that the training accuracy better than validation accuracy which implies model can predict training data better, the training curve is not noisy and validation curve is noisy.

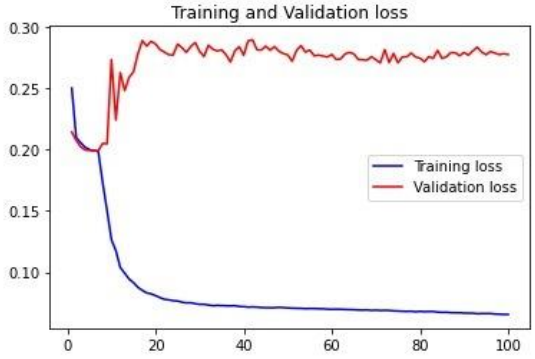

Figure 13: Graph plotted between Training loss vs Validation loss

From figure 13, we infer that these plots and graphs, we infer that the model is trained to a good extent as the curve does not keep decreasing and thus fine tuning hyperparameters beyond this point is not required and the training set is much easier to predict than validation set.

Here, our work is to detect the anomaly in videos. An anomaly detection is a term used widely in the field of deep learning and computer vision. And there are a lot of techniques to detect anomalies. Like, multiple instance learning, station temporal method, autoencoders, gaussian analysis are the most frequently used techniques to detect anomaly in videos. Multiple instance learning is the latest technique used to detect anomaly videos, and in multiple instances learning even the anomaly videos are given into the training set and if any one frame in a video is labelled anomaly, then the whole video is placed in the anomaly basket and then they are fed to the model, but the disadvantage here is its testing accuracy. The dataset we used for our work was collected by Dr. Chen Chen and he has used multiple instances learning with two models named "C3D Feature Extraction" and "TCNN" and the accuracy of the accuracy of the model was 23.6 and 22.8 percentage respectively. But on the other hand, we have spatio-temporal methods with audio encoders embedded in it and even in such a method, we weren't able to see a drastic change in the accuracy of the model. Thus, we planned on inducing audio encoders with the gaussian analysis. Here we used an auto encoder mechanism to divide the video into frames and convert each frame to gray scale and then apply gaussian blur to reduce the noise in the frames and then give it in a model similar to gaussian analysis method. We give our denoised frames to a model named "LSTM" which analyzes the normal videos and forms a median curve and with loss for each frame calculated by calculating the squared mean difference which shows the difference from the mean of the curve. If the loss is too high then it means that the frames deviate from the pattern of the normal frames. And then the loss of the training set is calculated and if it's larger than the pattern of the loss of the normal frames, then it is considered an anomaly video and the accuracy of our data set is given below:

Avenue dataset - 74.56022620%

Thus, our accuracy is much increased when compared to the work done by Dr. Chen Chen who was the actual one who collected the anomaly dataset and now our work is open for future research purposes. And ours will set a benchmark for accuracy of detecting the anomaly in the “Anomaly detection dataset”.

Conclusion/Future work

The main aspect is using gray and gaussian blur to denoise the frames. Denoising is the technique of removing noise or distortions from a frame. There are a vast range of applications such as blurred images that can be made clear. A gray scale image consists of colors which are shades of gray. The advantage of a gray scale image over other images is that each pixel is provided with very less information, thus the size of the image reduces and the computational time decreases. In a Gaussian blur, more weightage is given to pixels which are close to center of the kernel.The filtered pixel is the average channel values of the old pixels where this averaging is done in a channel-by-channel basis. Fine-tuning the hyper-parameters such as epochs, batch_size, patience, steps_per_epoch. From the table, we were able to observe that as the epochs increased the accuracy was constant. But along with the epochs and steps_per_epoch increases the accuracy has increased. But whereas the patience changing the patience does not have a great impact

on increasing the accuracy of the model. So, these are the critical aspects of our work.

The main aspect is using gray and gaussian blur to denoise the frames. Denoising is that the technique of removing noise or distortions from a frame. there's an unlimited range of applications like blurred images that may be made clear. A grayscale image is one where all the colors present in it are purely shades of gray. the explanation for differentiating such images from the other type of color image is that less information has to be provided for every pixel. Fine-tuning the hyper-parameters like epochs, batch_size, patience, steps_per_epoch. From the table, we were ready to observe that because the epochs increased the accuracy was constant. Along with the epochs, and steps_per_epoch the accuracy has increased. But whereas the patience changing the patience doesn't have an excellent impact on increasing the accuracy of the model. So, these are the critical aspects of our work.

We propose a model with 3D Convolution networks that helps identify and predict anomalies in videos. The data fed in this model is de-noised with the help of Gaussian blur technique. This helps improve the efficiency of data fed into the training the model.

Future work includes investigating other network architecture. Dimensionality reduction can be done to improve accuracy. When we try to plot the training accuracy vs validation loss the curve descends indicating that the model is open to more training. We plan to better the model by trying other techniques and finally applying our model to more complex situations and see how it performs.

References

[1] Guo, J., Zheng, P. and Huang, J., 2019. Efficient privacy-preserving anomaly detection and localization in bitstream video. IEEE Transactions on Circuits and Systems for Video Technology, 30(9), pp.3268-3281.

[2] Zhou, J.T., Du, J., Zhu, H., Peng, X., Liu, Y. and Goh, R.S.M., 2019. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, *14*(10), pp.2537-2550.

[3] Luo, W., Liu, W., Lian, D., Tang, J., Duan, L., Peng, X. and Gao, S., 2019. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE transactions on pattern analysis and machine intelligence*.

[4] Cheng, K.W., Chen, Y.T. and Fang, W.H., 2015. Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation. *IEEE Transactions on Image Processing*, *24*(12), pp.5288-5301.

[5] Chu, W., Xue, H., Yao, C. and Cai, D., 2018. Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos. *IEEE Transactions on Multimedia*, *21*(1), pp.246-255.

[6] Li, Y., Cai, Y., Liu, J., Lang, S. and Zhang, X., 2019. Spatio-temporal unity networking for video anomaly detection. *IEEE Access*, *7*, pp.172425-172432.

[7] Leyva, R., Sanchez, V. and Li, C.T., 2017. Video anomaly detection with compact feature sets for online performance. *IEEE Transactions on Image Processing*, *26*(7), pp.3463-3478.

[8] Ganokratanaa, T., Aramvith, S. and Sebe, N., 2020. Unsupervised anomaly detection and localization based on deep spatiotemporal translation network. *IEEE Access*, *8*, pp.50312-50329.

[9] Wang, S., Zeng, Y., Liu, Q., Zhu, C., Zhu, E. and Yin, J., 2018, October. Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection. In *Proceedings of the 26th ACM international conference on Multimedia* (pp. 636-644).

[10] Jardim, E., Thomaz, L.A., da Silva, E.A. and Netto, S.L., 2019. Domain-transformable sparse representation for anomaly detection in moving-camera videos. *IEEE Transactions on Image Processing*, *29*, pp.1329-1343.

[11] Li, N., Chang, F. and Liu, C., 2020. Spatial-temporal cascade autoencoder for video anomaly detection in crowded scenes. *IEEE Transactions on Multimedia*, *23*, pp.203-215.

[12] Nawaratne, R., Alahakoon, D., De Silva, D. and Yu, X., 2019. Spatiotemporal anomaly detection using deep learning for real-time video surveillance. *IEEE Transactions on Industrial Informatics*, *16*(1), pp.393-402.

[13] Xiang, T. and Gong, S., 2008. Video behavior profiling for anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, *30*(5), pp.893-908.

[14] Thomaz, L.A., Jardim, E., da Silva, A.F., da Silva, E.A., Netto, S.L. and Krim, H., 2017. Anomaly detection in moving-camera video sequences using principal subspace analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *65*(3), pp.1003-1015.

[15] Li, W., Mahadevan, V. and Vasconcelos, N., 2013. Anomaly detection and localization in crowded scenes. *IEEE transactions on pattern analysis and machine intelligence*, *36*(1), pp.18-32.

[16] Xu, K., Jiang, X. and Sun, T., 2018. Anomaly detection based on stacked sparse coding with intraframe classification strategy. *IEEE Transactions on Multimedia*, *20*(5), pp.1062-1074.

[17] Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H. and Hua, X.S., 2017, October. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia* (pp. 1933-1941).

[18] Yuan, Y., Fang, J. and Wang, Q., 2014. Online anomaly detection in crowd scenes via structure analysis. *IEEE transactions on cybernetics*, *45*(3), pp.548-561.

[19] Tariq, S., Farooq, H., Jaleel, A. and Wasif, S.M., 2021. Anomaly detection with particle filtering for online video surveillance. *IEEE Access*, *9*, pp.19457-19468.

Github:

https://github.com/sneha-da/anomalyDetection