Detecting Anomalies in Videos Using Reconstruction and Prediction based Deep Learning Approach

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Niraj Kumar

[Roll No: CS-2011]

under the guidance of

Dr. Ashish Ghosh.
Professor
Machine Intelligence Unit

Indian Statistical Institute
Kolkata-700108, India

July 2022

CERTIFICATE

This is to certify that the dissertation entitled **"Detecting Anomalies in Videos Using Reconstruction and Prediction Based Deep Learning Approach"** submitted by **Niraj Kumar** to Indian Statistical Institute, Kolkata, in partial fulfillment for the award of the degree of **Master of Technology in Computer Science** is a bonafide record of work carried out by him under my supervision and guidance. The dissertation has fulfilled all the requirements as per the regulations of this institute.

Dr. Ashish Ghosh
Professor,
Machine Intelligence Unit,
Indian Statistical Institute,
Kolkata-700108, INDIA.

Acknowledgements

I wish to express my sincere appreciation to my supervisor, *Prof. Ashish Ghosh*, Machine Intelligence Unit, Indian Statistical Institute, Kolkata, for his guidance and continuous support and encouragement. Special thanks go the Machine Intelligence Unit(MIU) of ISI Kolkata for providing technical support, and other essential resources. Senior research scholars specially *Subhadip Boral*, and *Anwesha Law* gave me quality of time from their busy academic schedule so that I can complete the dissertation conveniently on time. Last but not the least, I want to thank my parent, who constantly encouraged me to do my best, and motivated me even in my bad days.

Niraj Kumar
CS2011
M.Tech in Computer Science,
Indian Statistical Institute
Kolkata - 700108 , India.

Abstract

Anomaly detection in videos deals with pointing out the events that are out of normal. Current methods deals with identification of anomalous frames in a video sequence based on certain objects, and behaviours present in the video. Anomalies in videos are continuous events, and due to high number of features, generally the classical methods are not good enough for the task. Most of the reconstruction based deep learning methods works on the assumption that anomalies are rare in nature, and the training sets doesn't contain any kind of anomalous events. This may work in case of object related anomalies, but will fail in case of motion related anomalies. We design a two-branch reconstruction and prediction based convolutional auto-encoder which utilises future frame prediction technique along with 3D convolutions to capture both spatial and temporal features. Moreover, the use of skip connections have been utilised in prediction branch to avoid the loss of spatial information during prediction in crowded frames. To overcome the problem of small dataset, we created new dataset by superimposing images over one another. This led to more data as well as frames containing more crowd density.
Keywords: *Anomalies* ; *spatial information*; *two-branch reconstruction*; *temporal features*; *3D convolution*; *convolutional autoencoder*

Contents

Chapter 1

Introduction

Anomaly detection in videos deals with identifying the events that are out of normal behaviour. With the increase in number of surveillance cameras, its impossible to have dedicated people for the surveillance. This led to development of anomaly detection algorithms. This task is extremely challenging because abnormal events are very rare in real world, some even might not have occurred before. So, normal classification based methods don't work in this case. There have been relevant work using classical methods, but as videos are high dimensional data, it's difficult to extract quality features using hand-crafted feature based methods. To counter this, various deep learning based techniques have been proposed. Reconstruction based approaches using deep learning are the most popular ones, but they might work in case of object based anomalies, but in case of motion based anomalies they don't seem to work very well. Moreover, the current methods are very peculiar to data-sets they are trained on, and well generalized methods are still not available as of now.

We designed a reconstruction and prediction based deep learning that tries to predict future frames in order to capture temporal features. Moreover, skip connections were used in the prediction branch, to recover the lost spatial information. Also, new small data-set was created by superimposing the images to create more crowded scenes.

Section 2 contains a brief discussion about the relevant past work. Section 3 deals with the preliminaries, and the base model we've tried improving. The data-sets used are discussed in section 4. Section 5 presents the proposed methodology and the intuition behind it. The experimental observations are reported in Section 6. In section 7, we have discussed about the pros and cons of our method, and in section 8, we discuss the further opportunities for the improvements. Section 9 contains all the relevant references used.

Chapter 2

Related Work

Based on the past work done in the field of video-anomaly detection, the related work can be classified into two categories mainly work related to Anomaly Detection and past work which have used 3D Convolution neural networks.

2.1 Anomaly Detection

Most of the video anomaly detection techniques train the model based on some local hand-crafted features extracted before the training.Cong et al.[4] uses sparse reconstruction cost with a normal dictionary for measuring the regularity score of the test-set based on Multi-Scale Histogram of Optical Flow(MHOF). Due to the limited representation of hand-creafted features, they are not able to handle complex video anomaly detection tasks.
Unlike traditional methods, deep learning based methods are able to learn high-dimensional features of videos and have proven to generate quite good results in many Computer vision tasks. Hasan et al.[5] proposed a fully-convolutional autoencoder for learning spatio-temporal features. This model was only able to extract spatial-features because of the convolution operation being performed in 2-dimensions only, eventhough multiple frames were given as input to the model. Xu et al.[16] desigend a stacked-autoencoder to extract the features, and later utilised a one-class SVM to separate out the anomalies. Patches are extracted from the frames and then passed to a fully-connected autoencoder after flattening. As the fully-connected autoencoders are permutation invariant, the spatial information was lost.

2.2 3D Convolutional Neural Networks

[14] shows that 3D convolution are well suited for temporal feature based tasks like action recognition. Some early works like [7] uses 3D convolution to design a neu-

ral network to learn spatio-temporal features in videos. Tran et al.[15] used a large video dataset [8] to train a deep 3D convolutional network, and was able to beat state-of-the-art models in action recognition. Varol et al. designed a long-term temporal convolution network using 3D convolutional to enhance the original models. Moreover, there have been already quite relevant papers recommending to used 3D convolution over 2D convolution for video related tasks. Also, due to the large imbalance in the dataset, classical binary classification based approaches, and typical deep neural network models can't be used.

Chapter 3

Preliminaries

3.1 Convolutional AutoEncoders

Convolution Autoencoders makes use of convolution to decompose the input signal as the sum of other signals. For an input image with single channel, the latent form of the i^{th} filter will be:

$$h^i = f(x + W^i) + b^i \tag{3.1}$$

Here, h^i is the latent form, f is the activation function, x is the input, W^i are the weights of the kernel, and b^i is the bias.

The encoder part in the convolutional autoencoders encodes the image in a latent representation, and later on the decoder part tries to reconstruct the desired output using the latent form.

Traditional Autoencoders, and PCA are permutation invariant i.e they ignore the spatial structure in the images. Moreover, they tend to have large redundancy in the network parameters, whereas Convolutional autoencoders tend to have fewer parameters due to weight sharing.

3.2 3D Convolutional Neural Networks

3D convolutional networks operates by performing convolution operation on a 3D volume using a 3D kernel, unlike 2D convolution which uses a 2D kernel and a 2D input.
3D ConvNets have better ability to capture temporal information from the input volume compared to 2D ConvNets, because of 3D convolution and 3D pooling. [**?**]. In 3D ConvNets, the convolution and pooling operation are performed spatio-temporally while in 2D ConvNets they are performed spatially only. Output of a 3D convolution

is a volume instead of a 2D.
Fig. 3 represents a typical process for 3D convolution.

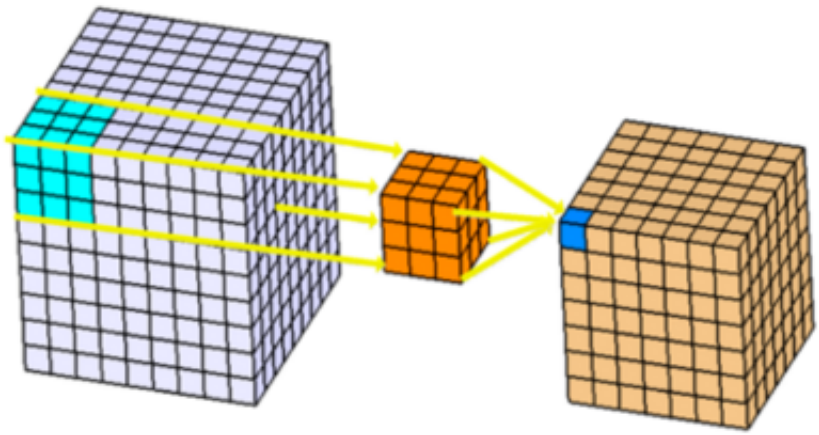

Figure 3.1: 3D Convolution[1]

3.3 Skip Connections

Skip connections skip some intermediate layers in a neural network to feed the output of one layer as input to other layers. Skip connections can be used either in form of addition, or either in form of concatenation. Addition is mostly used in residual architectures, while the concatenation is used in densely connected architectures.

Addition is mostly used to overcome the gradient vanishing problem, while concatenation is used for feature re-usability, and making models more compact. For feature re-usability, mostly long-skip connections are used instead of short ones.
Fig. 3.2 represents a typical example of skip connection.

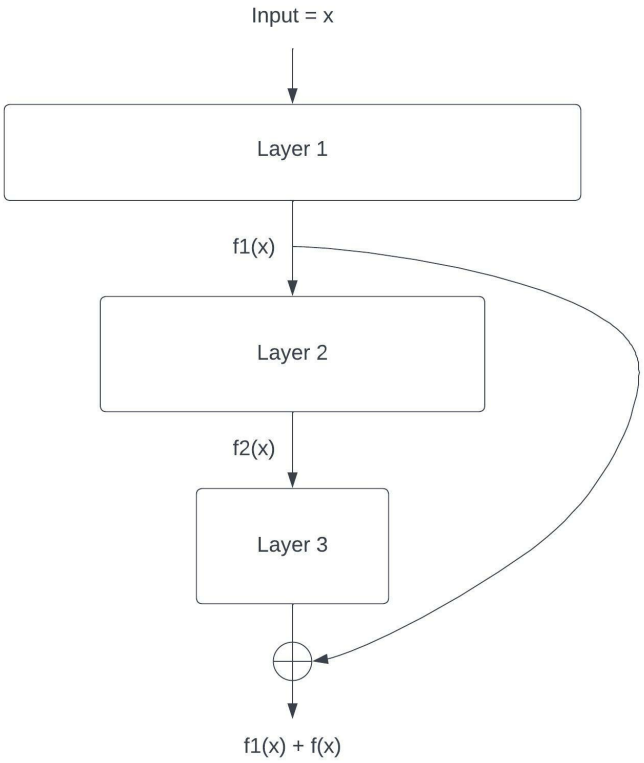

Figure 3.2: Skip Connection

Chapter 4

Datasets

4.1 UCSD Ped

UCSD Ped[2] contains two subsets, Ped1 and Ped2. Each subset is captured using different cameras.
Ped1 is captured using a static camera. It contains 34 short video clips as training set, and 36 clips as part of testing set. Each training video clip has around 200 frames with a resolution of 238x158 pixels.
Ped2 contains 16 short clips as part of training set, and another 12 clips in testing set. Each clip in training set has around 120 or 150 or 180 frames. Number of frames in testing videos also ranges between 120 to 180. Here also, each frame is captured by a static camera, and the resolution of each frame is 240x360 pixels.
The walking pedestrians are considered as normal, whereas bicycles, cars, carts or people using skateboards are considered as anomaly. Moreover, unusual motion by pedestrians is also considered as anomalous. Some frames contain high crowd density making the dataset a bit more challenging.

(a) From Ped1

(b) From Ped2

(c) From Ped2

Figure 4.1: Sample Images from the datasets

Video clip	No. of frames	Anomaly Type	Anomalous Frames
1	200	Bicycle	60 - 152
2	200	Bicycle, Skateboard	50 - 175
3	200	Bicycle	91 - 200
4	200	Skateboard	31 - 168
5	200	Bicycle	5 - 90, 140 - 200
6	200	Bicycle, Skateboard	1 - 100, 110 - 200
7	200	Bicycle, Skateboard	1 - 175
8	200	Skateboard	1 - 94
9	200	Walking on the grass	1 - 48
10	200	Skateboard	1 - 140
11	200	Walking on the grass	70 - 165
12	200	Skateboard	130 - 200
13	200	Trolley	1 - 156
14	200	Bicycle, Truck	1 - 200
15	200	Bicycle	138 - 200
16	200	Bicycle, Walking on the grass	123 - 200
17	200	Bicycle	1 - 47
18	200	Skateboard	54 - 120
19	200	Van	64 - 138
20	200	Truck	45 - 175
21	200	Bike	31 - 200
22	200	Skateboard	16 - 107
23	200	Bike, Skateboard	8 - 165
24	200	Cart, Skateboard	50 - 171
25	200	Skateboard	40 - 135
26	200	Bicycle	77 - 144
27	200	Truck	10 - 122
28	200	Bicycle	105 - 200
29	200	Bicycle	1 - 15, 45 - 113
30	200	Bicycle	175 - 200
31	200	Bicycle, Walking on the grass	1 - 180
32	200	Bicycle	1 - 52, 65 - 115
33	200	Bicycle	5 - 165
34	200	Skateboard	1 - 121
35	200	Skateboard	86 - 200
36	200	Bicycle and Truck	15 - 108

Table 4.1: UCSD ped1 test set

Video clip	No. of frames	Anomaly Type	Anomalous Frames
1	180	Bicycles	61 - 180
2	180	Bicycles	95 - 180
3	150	Bicycles	1 - 146
4	180	Carts and Bicycles	31 - 180
5	150	Bicycles	1 - 129
6	180	Bicycles	1 - 159
7	180	Bicycles and Skateboards	46 - 180
8	180	Bicycles and Skateboards	1 - 180
9	120	Bicycles	1 - 120
10	150	Bicycles	1 - 150
11	180	Bicycles	1 - 180
12	180	Skateboards	88 - 180

Table 4.2: UCSD ped2 test set

Chapter 5

Proposed Methodology

5.1 Data Augmentation and Pre-processing

The training frames were pre-processed to create a volume of k consecutive frames by sliding a window of size k with stride of 1. For each volume, the next set of k consecutive frames were treated as the output for prediction branch.
Moreover, for each training set, after splitting it in two equal halves, consecutive images at same indices in both the halves were superimposed on each other to create more crowded scenes, so as to let the model learn how to extract features even in case of crowded videos.
Each test frame is labelled as 1 or 0. 1 for being anomalous, and 0 for being a normal frame. Similar to training data, volumes were created for testing data as well. Labelling was done for each volume. If at least 40 percent of the frame in a volume are anomalous, the the volume is treated as anomalous else it is treated as a normal volume.

(a) First Image (b) Second Image (c) Superimposed Image

Figure 5.1: Creating densely crowded images

5.2 Network Architecture

Instead of detecting anomality for each frame, we try to find whether a volume of frame is anomalous or not. As the anomalies are continuous in nature [11], we assume that concluding a random frame as anomalous from a video captured in high fps is not plausible. Instead a continuous volume of frames can be used to detect whether a part of video is anomalous or not. A 2-branch output model consisting of a reconstruction and prediction branch respectively, which is inspired from [18].
The volume size is set to $k = 8$, and each frame was resized to $64 \times 128 \times 1$. So, the input volume has the shape $8 \times 64 \times 128 \times 1$. The encoder part has four 3D convolution [14] layers to extract the spatio-temporal features from the input volume. We have used the kernel of $3 \times 3 \times 3$ with strides of $1 \times 1 \times 1$ in all the layers as suggested by Trans. et al [15]. The feature maps produced by each Convolution layer is a 3D tensor containing the temporal information as well. Number of kernels are first increased sequentially to encode the image in latent form, and in decoder part the kernels are used in symmetric manner to deconvolve the inputs. To accelerate the convergence during training, batch normalization [6] has been used. Leaky ReLU [10] is used as activation function in all the intermediate layers. 3D max-pool layers are used after the activation, with strides of $2 \times 2 \times 2$ and pool size of $2 \times 2 \times 2$. Gradient operation of convolution called Deconvolution operation [17] is used in the decoder part instead of 3D convolution. The last layer of the decoder part is a 3D convolution layer which tries to restore back the number of channels with respect to the input image. For input data normalization, Sigmoid activation function is being used at the last layer.

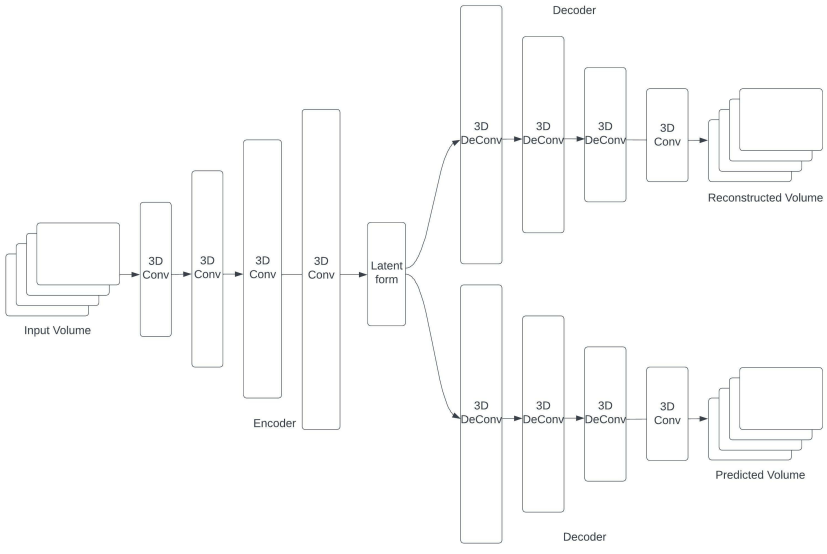

Figure 5.2: Base model Architecture

The improved version of the base model contains skip connections in the prediction branch, so as to reuse the feature maps from the common encoder part. It was inspired from other models using skip connections like U-net [12] etc. As the prediction branch deals with temporal features, it was favouring temporal features over the spatial ones, leading to poor reconstruction even in case of normal data. This was a major problem in crowded frames. Skip connections allows the decoder part to reuse the spatial features obtained during encoding leading to better reconstruction.
In this architecture, skip connections are used as long-skip connections, with concatenation operation. The concatenation happens in the last dimension of the tensors, concatenating the feature maps from encoder part to the decoder part.

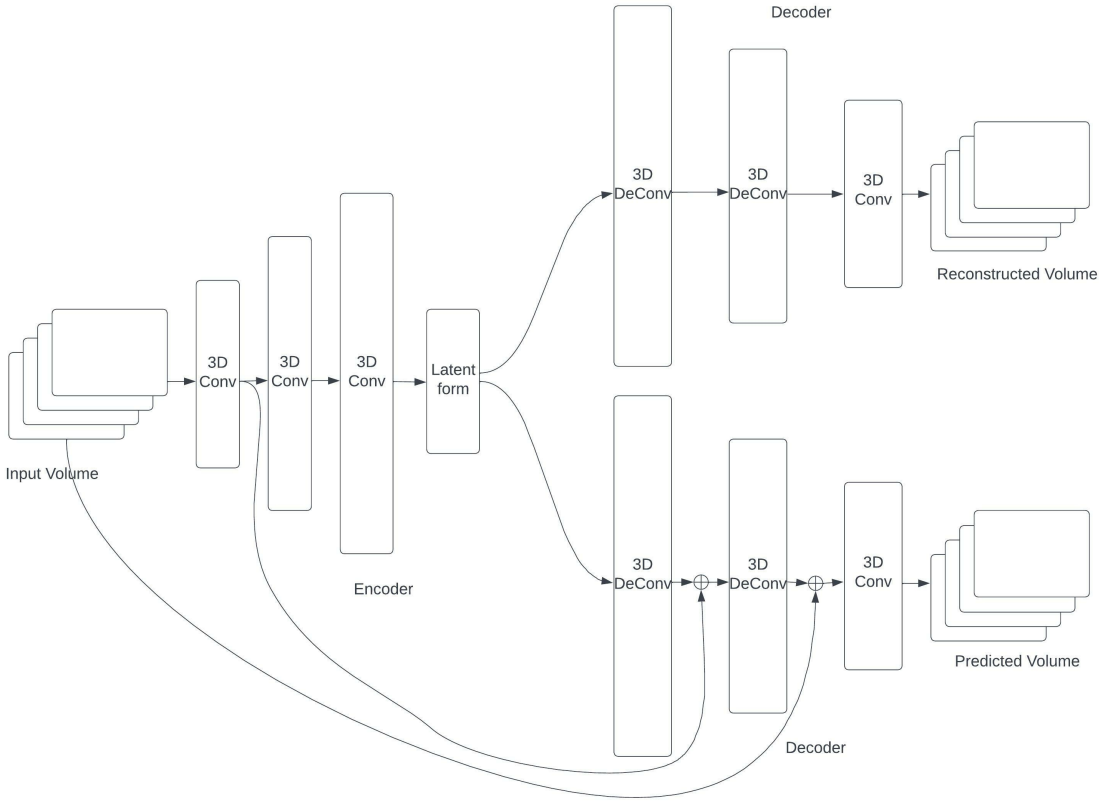

Figure 5.3: Base model with Skip connections

5.2.1 Objective Function and Loss Functions

The Reconstruction loss can be given by:

$$loss_{rec} = \frac{1}{N}\sum_{i=1}^{N} ||V_i - f_{rec}(V_i)||_2^2 \tag{5.1}$$

Here, N is the batch size, V_i is the i^{th} volume in the batch, and $f_{rec}(V_i)$ is the reconstructed i^{th} volume.

The prediction loss can be given as:

$$loss_{pred} = \frac{1}{N}\sum_{i=1}^{N} ||V_i - f_{pred}(V_i)||_2^2 \tag{5.2}$$

Here, N is the batch size, V_i is the i^{th} volume in the batch, and $f_{pred}(V_i)$ is the predicted $i + 1^{th}$ volume.

The optimization objective of the model is:

$$\min_{w} w_1 * loss_{rec} + w2 * loss_{pred} \tag{5.3}$$

Here, w is the parameters of the model, $loss_{rec}$ and $loss_{pred}$ are the reconstruction loss and prediction loss respectively, and w_1 and w_2 are the weights assigned to reconstruction loss and prediction loss respectively.

5.3 Evaluation criteria

The volume with error greater than a particular threshold are treated as abnormal volumes, while those with error lower than the threshold are treated as normal ones. Once, the errors for all the volumes from a particular video are calculated, the errors are normalized using the equation 5.4.

$$e(v) = \frac{e(v) - min_v e(v)}{max_v e(v) - min_v e(v)} \tag{5.4}$$

5.3.1 Abnormality Score

The abnormality score for a video sequence v can be given by:

$$a(v) = \frac{e(v) - min_v e(v)}{max_v e(v)} \tag{5.5}$$

Here, $e(v)$ is the error for volume v, $min_v e(v)$ is the minimum of all the reconstruction errors in the whole dataset, and $max_v e(v)$ is the maximum of all the reconstruction errors in the whole dataset.

Volume containing anomalous events will have higher value of $a(v)$ compared to a normal volume. In the real world, future data is not available, so $min_v e(v)$ and $max_v e(v)$ are set experimentally after the training.

Chapter 6

Experimental Results

All the three variations of the model was tested on Ped1 and Ped2. The results were calculated for each video to investigate the model's performance with respect to different types of anomalies, and later a combined result was calculated in terms of AUC score and EER score to get the overall idea of how the model performs on the whole data-set.

6.1 Base Model

6.1.1 Ped1

Figure 6.1 shows a frame containing a van. The corresponding heatmaps clearly shows the reconstruction error in blue color bounded in red rectangle. Figure 6.2 clearly shows that the model is not able to reconstruct a small but crowded group of people leading to a false alarm. Figure 6.3 shows a skateboarder being captured during the reconstruction as well as prediction phase.
Figure 6.4 shows the AUC-ROC curve for the model with a score of 0.768. Table 6.1 shows the false alarms, missed alarms, AUC, and EER for each video respectively.

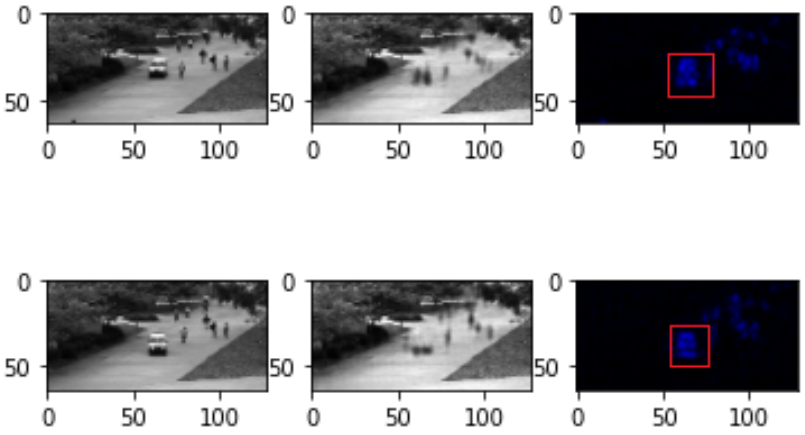

Figure 6.1: Sample from Testing video 19: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

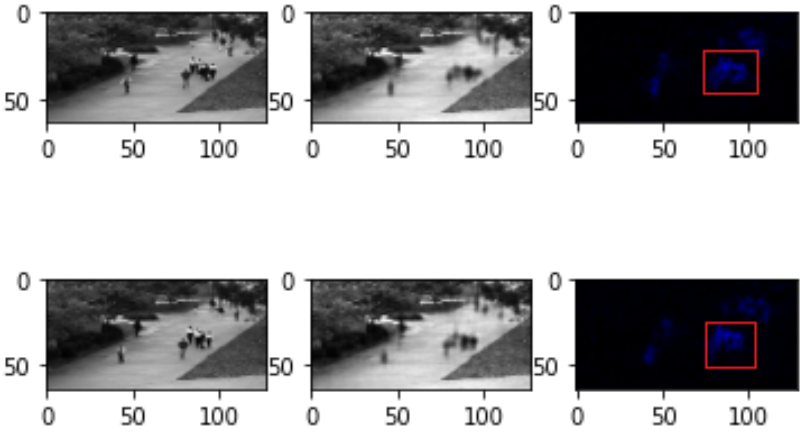

Figure 6.2: Sample from Testing video 34: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycle	25	0	0.97	0.098
2	Bicycle, Skateboard	12	1	0.99	0.015
3	Bicycle	15	0	0.99	0.043
4	Skateboard	26	42	0.46	0.609
5	Bicycle	28	33	0.61	0.36
6	Bicycle, Skateboard	0	45	0.99	0.02
7	Bicycle, Skateboard	0	5	0.99	0.01
8	Skateboard	46	20	0.72	0.37
9	Walking on the grass	70	0	0.99	0.04
10	Skateboard	0	22	0.97	0.05
11	Walking on the grass	22	78	0.37	0.64
12	Skateboard	8	18	0.79	0.23
13	Trolley	0	10	0.98	0.06
14	Bicycle, Truck	0	63	-	-
15	Bicycle	49	2	0.89	0.19
16	Bicycle, Walking on the grass	77	0	0.87	0.23
17	Bicycle	80	34	0.23	0.63
18	Skateboard	81	41	0.30	0.67
19	Van	29	2	0.98	0.02
20	Truck	0	38	0.92	0.21
21	Bike	20	47	0.59	0.38
22	Skateboard	29	17	0.69	0.33
23	Bike, Skateboard	19	39	0.56	0.40
24	Cart, Skateboard	27	1	0.99	0.001
25	Skateboard	5	10	0.97	0.09
26	Bicycle	25	15	0.74	0.27
27	Truck	0	39	0.99	0.004
28	Bicycle	70	15	0.42	0.61
29	Bicycle	8	31	0.91	0.21
30	Bicycle	8	0	1.0	0.0
31	Bicycle, Walking on the grass	0	21	0.98	0.14
32	Bicycle	1	0	0.99	0.03
33	Bicycle	0	20	1.0	0.0
34	Skateboard	66	60	0.11	0.81
35	Skateboard	28	34	0.76	0.33
36	Bicycle and Truck	0	26	0.90	0.16

Table 6.1: UCSD ped1 results

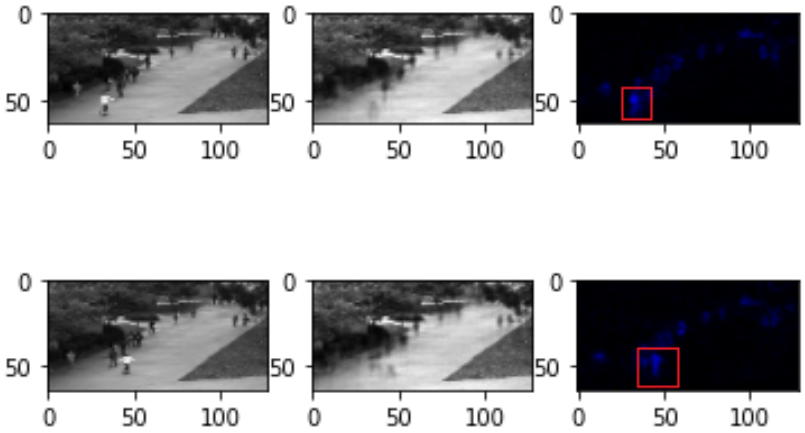

Figure 6.3: Sample from Testing video 35: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

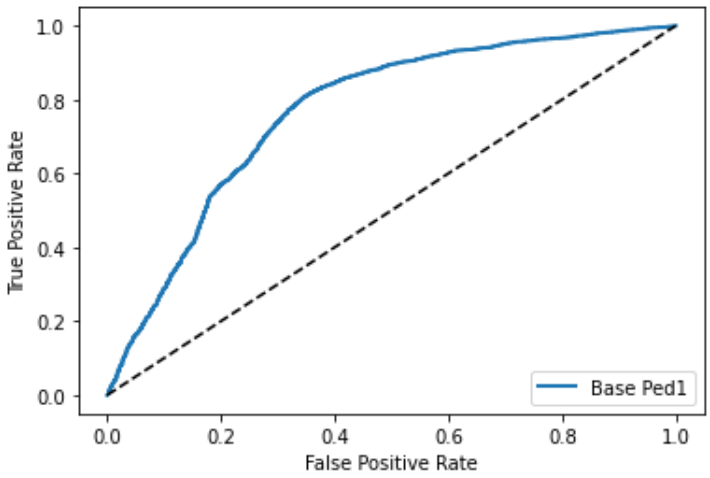

Figure 6.4: AUC-ROC curve for ped1

6.1.2 Ped2

Figure 6.5, and 6.6 shows poor reconstruction as well poor prediction leading to high reconstruction for crowd in the heatmaps.
Figure 6.7 shows the AUC-ROC curve for the model with a score of 0.777. Table 6.2 shows the false alarms, missed alarms, AUC, and EER for each video respectively.

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycles	56	28	0.26	0.63
2	Bicycles	1	0	1.0	0.0
3	Bicycles	0	10	-	-
4	Carts and Bicycles	0	6	1.0	0.0
5	Bicycles	0	34	0.89	0.22
6	Bicycles	7	22	0.83	0.09
7	Bicycles and Skateboards	2	0	1.0	0.0
8	Bicycles and Skateboards	0	29	-	-
9	Bicycles	0	17	-	-
10	Bicycles	0	72	-	-
11	Bicycles	0	57	-	-
12	Skateboards	44	4	0.80	0.28

Table 6.2: UCSD ped2 results

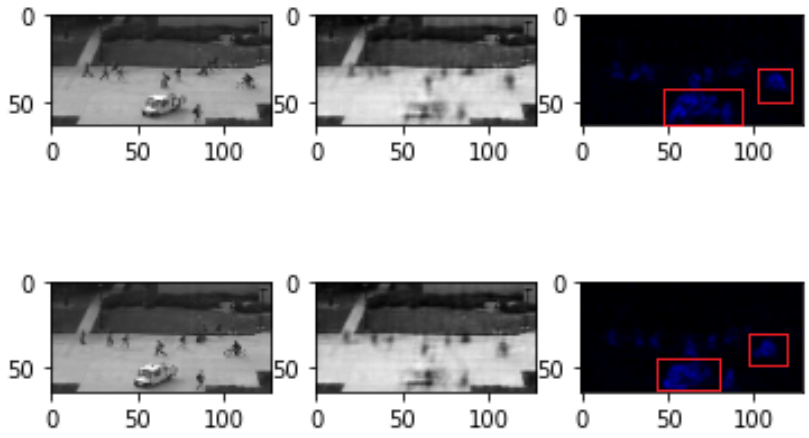

Figure 6.5: Sample from Testing video 4: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

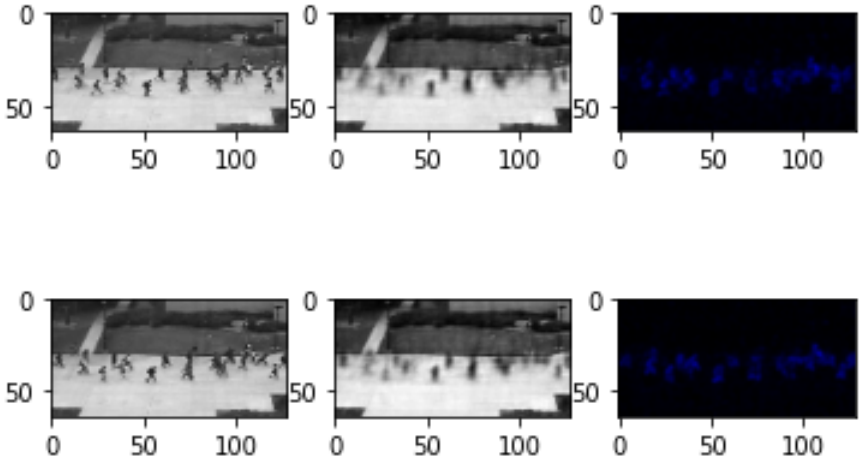

Figure 6.6: Sample from Testing video 12: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

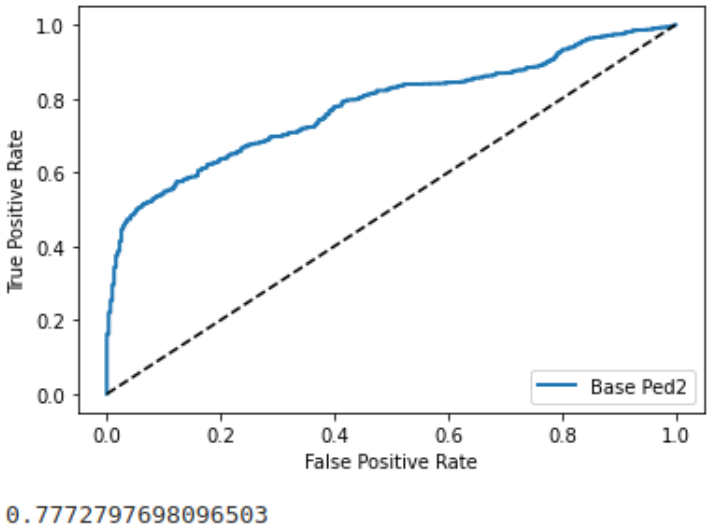

Figure 6.7: AUC-ROC curve for Ped2

6.2 Base Model with Skip connections in Prediction branch

6.2.1 Ped1

Figure 6.8 shows a frame containing a van. The corresponding heatmaps clearly shows the reconstruction error in blue color bounded in red rectangle. Because the anomalous object is big enough, there seems to be no difference from the output produced by the base model. Heatmaps in Figure 6.9 clearly shows that the model is able to reduce the reconstruction error for the small but crowded group of people preventing a false alarm. Figure 6.10 shows a skateboarder being reconstructed well, but still the reconstruction error seems high maybe because of shifting of pixels due to temporal information.
Figure 6.11 shows the AUC-ROC curve for the model with a score of 0.757. Table 6.3 shows the false alarms, missed alarms, AUC, and EER for each video respectively

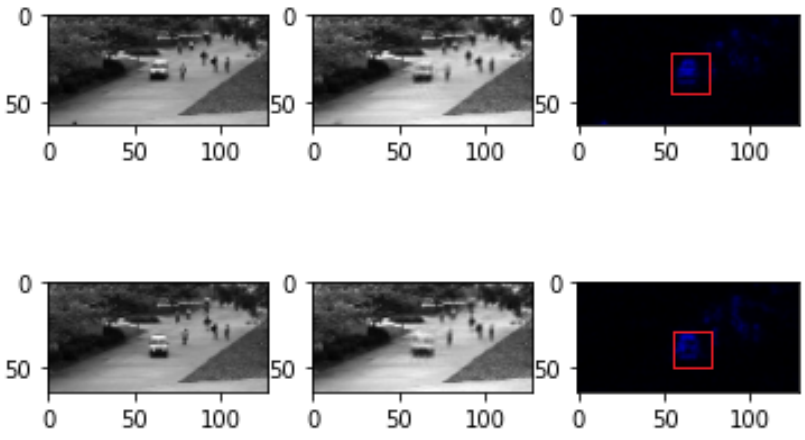

Figure 6.8: Sample from Testing video 19: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycle	36	1	0.99	0.01
2	Bicycle, Skateboard	22	0	0.99	0.01
3	Bicycle	13	6	0.97	0.09
4	Skateboard	26	49	0.41	0.62
5	Bicycle	36	31	0.67	0.29
6	Bicycle, Skateboard	0	19	0.99	0.008
7	Bicycle, Skateboard	0	7	0.99	0.01
8	Skateboard	28	0	0.98	0.06
9	Walking on the grass	89	0	0.98	0.06
10	Skateboard	0	11	0.98	0.07
11	Walking on the grass	31	71	0.43	0.55
12	Skateboard	8	59	0.76	0.19
13	Trolley	0	7	0.98	0.01
14	Bicycle, Truck	0	69	-	-
15	Bicycle	61	0	0.90	0.14
16	Bicycle, Walking on the grass	85	0	0.83	0.23
17	Bicycle	80	24	0.28	0.58
18	Skateboard	67	17	0.44	0.51
19	Van	8	2	0.93	0.05
20	Truck	0	34	0.92	0.23
21	Bike	26	20	0.73	0.27
22	Skateboard	47	1	0.68	0.33
23	Bike, Skateboard	25	8	0.28	0.62
24	Cart, Skateboard	10	2	0.98	0.04
25	Skateboard	4	16	0.93	0.13
26	Bicycle	63	10	0.82	0.21
27	Truck	0	33	0.98	0.05
28	Bicycle	68	35	0.33	0.66
29	Bicycle	60	14	0.73	0.38
30	Bicycle	23	0	1.0	0.0
31	Bicycle, Walking on the grass	0	41	0.99	0.005
32	Bicycle	14	3	0.98	0.10
33	Bicycle	0	31	0.99	0.003
34	Skateboard	64	38	0.35	0.63
35	Skateboard	16	42	0.82	0.25
36	Bicycle and Truck	0	25	0.87	0.16

Table 6.3: UCSD ped1 results

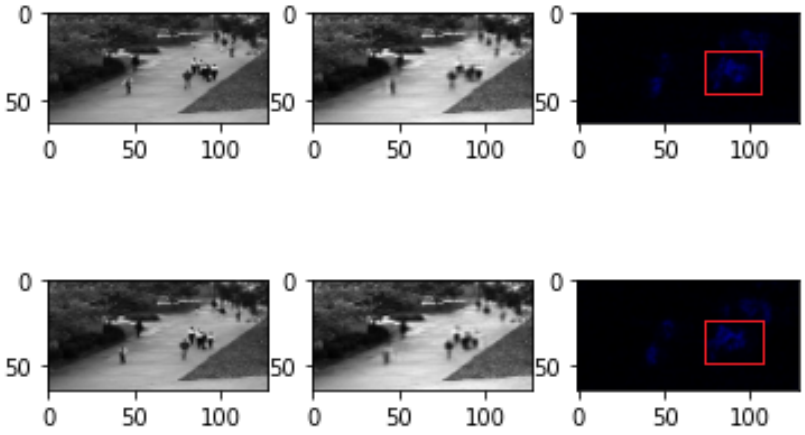

Figure 6.9: Sample from Testing video 34: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

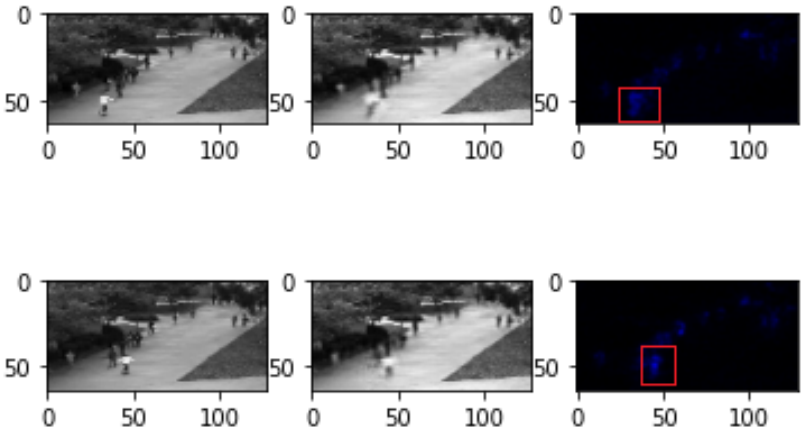

Figure 6.10: Sample from Testing video 35: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

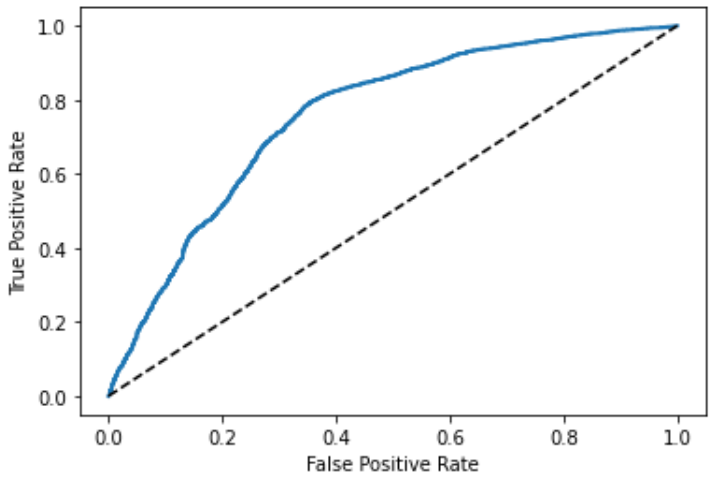

Figure 6.11: AUC-ROC curve for ped1

6.2.2 Ped2

Figure 6.12 shows high reconstruction error for both the bicyle and the truck in the heatmap, compared to the crowd. Figure 6.13 also shows fairly well reconstruction leading to low reconstruction error in heatmap.
Figure 6.14 shows the AUC-ROC curve for the model with a score of 0.796, and Table 6.4 shows the false alarms, missed alarms, AUC, and EER for each video respectively.

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycles	56	53	0.16	0.78
2	Bicycles	0	0	1.0	0.0.
3	Bicycles	0	14	-	-
4	Carts and Bicycles	0	7	1.0	0.0
5	Bicycles	0	16	0.99	0.01
6	Bicycles	0	31	0.83	0.22
7	Bicycles and Skateboards	4	0	1.0	0.0
8	Bicycles and Skateboards	0	29	-	-
9	Bicycles	0	29	-	-
10	Bicycles	0	71	-	-
11	Bicycles	0	76	-	-
12	Skateboards	37	4	0.91	0.17

Table 6.4: UCSD ped2 results

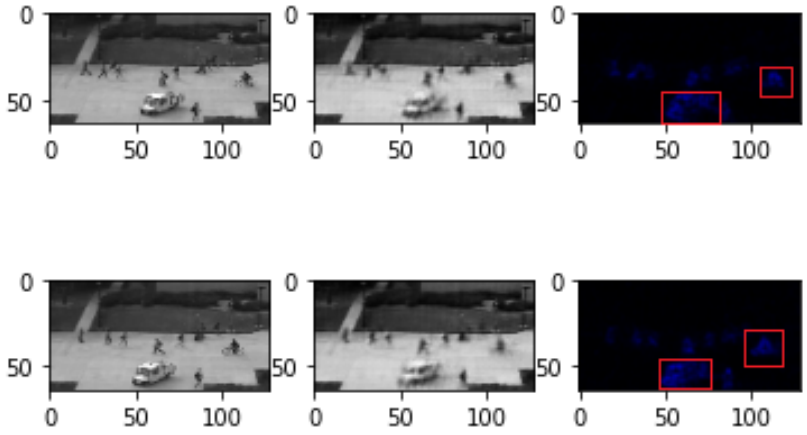

Figure 6.12: Sample from Testing video 4: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

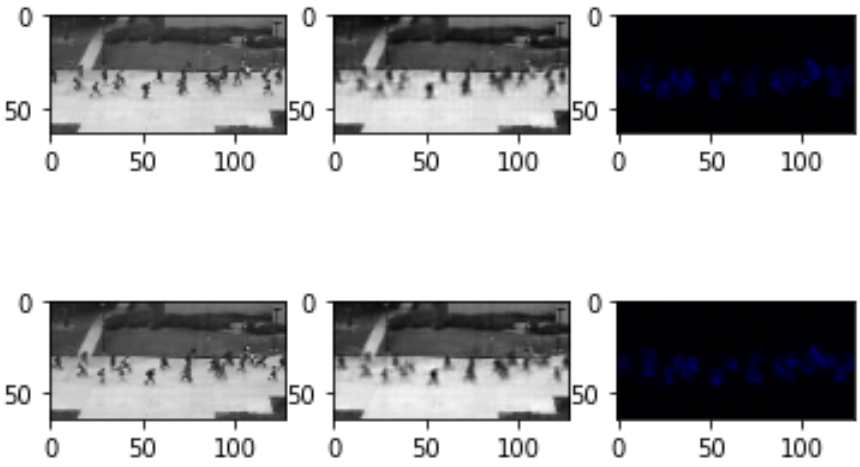

Figure 6.13: Sample from Testing video 12: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

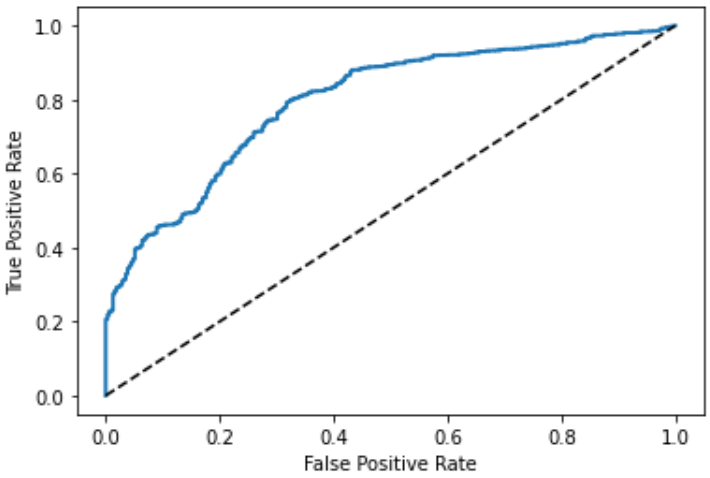

Figure 6.14: AUC-ROC curve for Ped2

6.3 Base Model with Skip connections and Data Augmentation

6.3.1 Ped1

Figure 6.15, 6.16, and 6.16 shows, there doesn't seems to be much difference between the images produced by the base model with skip connections and this model.
Figure 6.18 shows the AUC-ROC curve for the model with a score of 0.768, and Table 6.5 shows the false alarms, missed alarms, AUC, and EER for each video respectively.

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycle	36	0	0.99	0.01
2	Bicycle, Skateboard	16	0	0.99	0.004
3	Bicycle	9	5	0.98	0.09
4	Skateboard	26	55	0.40	0.68
5	Bicycle	30	31	0.65	0.32
6	Bicycle, Skateboard	0	34	0.99	0.002
7	Bicycle, Skateboard	0	21	0.99	0.008
8	Skateboard	15	0	0.97	0.08
9	Walking on the grass	78	0	0.98	0.10
10	Skateboard	0	13	0.98	0.04
11	Walking on the grass	29	73	0.42	0.54
12	Skateboard	8	59	0.76	0.20
13	Trolley	0	7	0.99	0.02
14	Bicycle, Truck	0	81	-	-
15	Bicycle	56	0	0.89	0.19
16	Bicycle, Walking on the grass	84	0	0.77	0.30
17	Bicycle	72	25	0.34	0.54
18	Skateboard	65	52	0.40	0.58
19	Van	8	3	0.93	0.05
20	Truck	0	43	0.91	0.25
21	Bike	24	25	0.78	0.22
22	Skateboard	45	1	0.69	0.30
23	Bike, Skateboard	25	37	0.37	0.56
24	Cart, Skateboard	8	2	0.99	0.04
25	Skateboard	27	6	0.96	0.10
26	Bicycle	34	18	0.77	0.28
27	Truck	0	59	0.99	0.04
28	Bicycle	75	1	0.40	0.57
29	Bicycle	34	19	0.81	0.27
30	Bicycle	13	0	1.0	0.0
31	Bicycle, Walking on the grass	0	42	0.99	0.01
32	Bicycle	12	2	0.99	0.04
33	Bicycle	0	41	1.0	0.0
34	Skateboard	63	39	0.40	0.55
35	Skateboard	14	52	0.79	0.33
36	Bicycle and Truck	0	24	0.90	0.14

Table 6.5: UCSD ped1 results

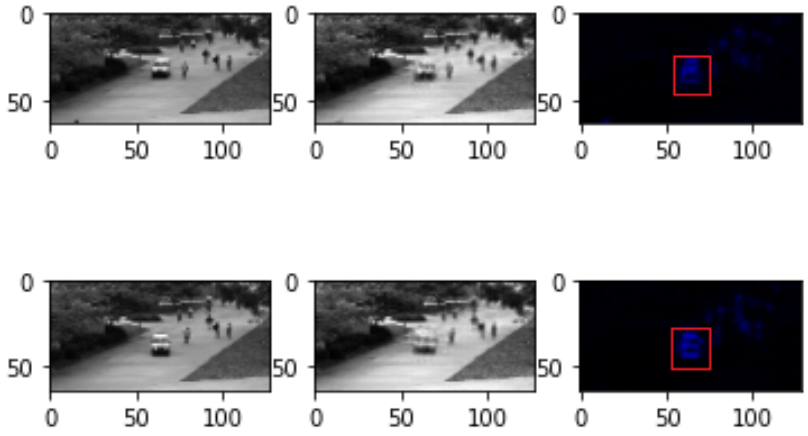

Figure 6.15: Sample from Testing video 19: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

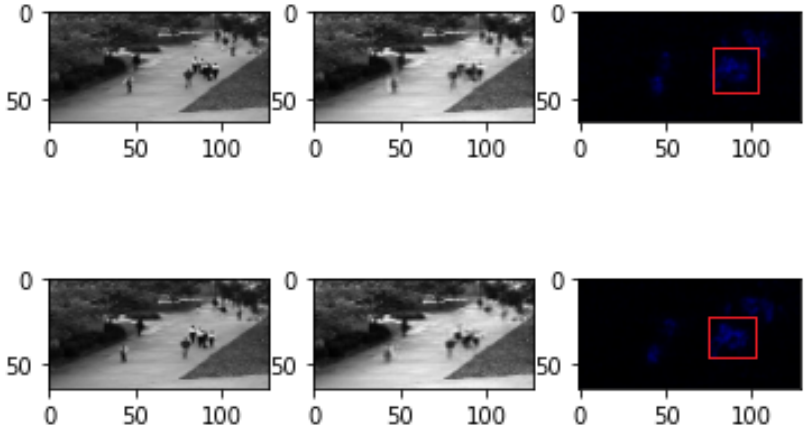

Figure 6.16: Sample from Testing video 34: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

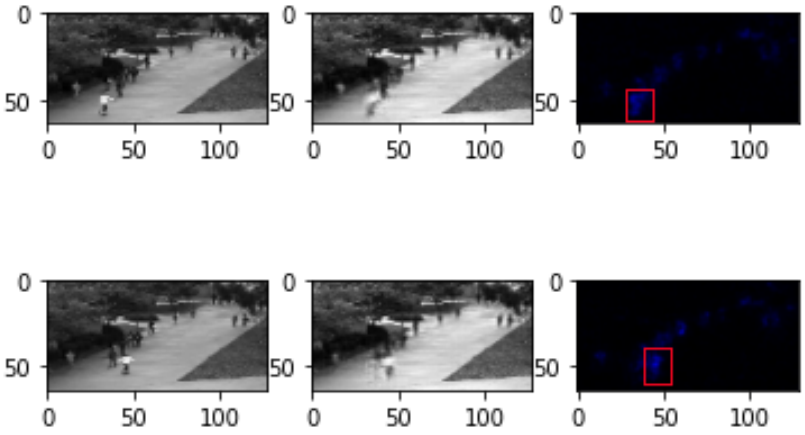

Figure 6.17: Sample from Testing video 35: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

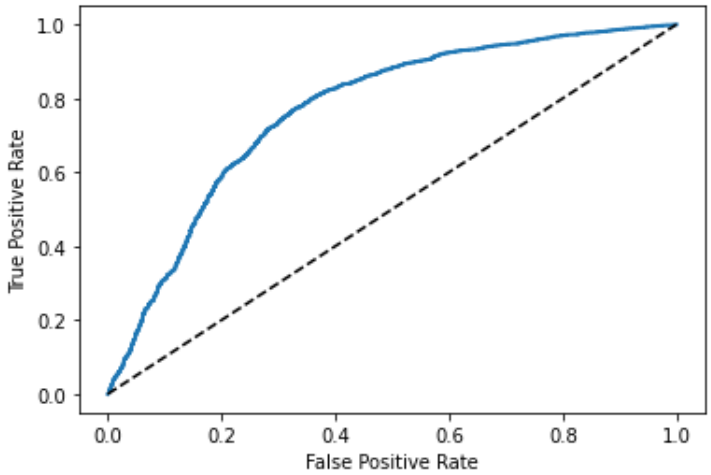

Figure 6.18: AUC-ROC curve for ped1

6.3.2 Ped2

Figure 6.19, and 6.20 clearly shows that the crowded scenes are reconstructed well, leading to decrease in number of false alarms. The result seems to be similar to base model with skip connections.
Figure 6.21 shows the AUC-ROC curve for the model with a score of 0.823, and Table 6.6 shows the false alarms, missed alarms, AUC, and EER for each video respectively.

Video clip	Anomaly Type	False Alarms	Missed Alarms	AUC	EER
1	Bicycles	41	57	0.33	0.66
2	Bicycles	0	0	1.0	0.0
3	Bicycles	0	15	-	-
4	Carts and Bicycles	0	10	1.0	0.0
5	Bicycles	0	15	1.0	0.0
6	Bicycles	0	33	0.83	0.09
7	Bicycles and Skateboards	0	1	1.0	0.0
8	Bicycles and Skateboards	0	37	-	-
9	Bicycles	0	29	-	-
10	Bicycles	0	68	-	-
11	Bicycles	0	89	-	-
12	Skateboards	39	4	0.90	0.18

Table 6.6: UCSD ped2 results

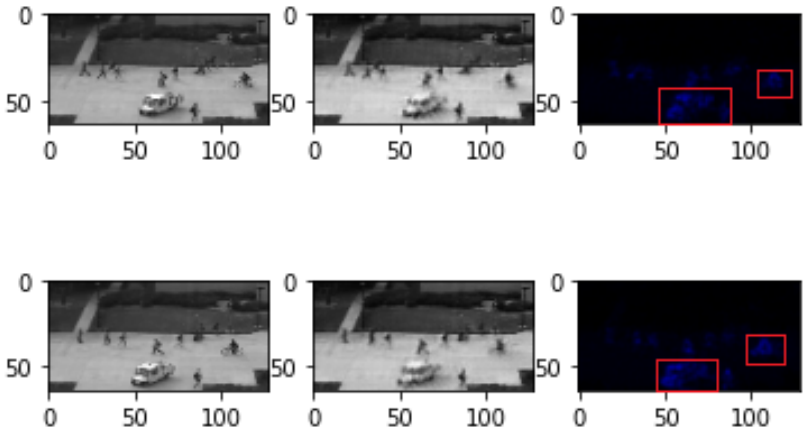

Figure 6.19: Sample from Testing video 4: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

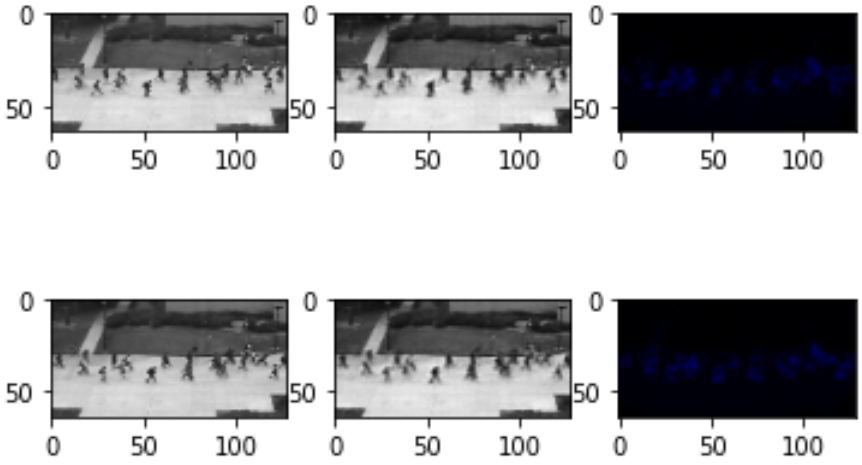

Figure 6.20: Sample from Testing video 12: Upper half contains the reconstruction phase, and lower half contains the prediction phase. First column contains ground truth images, while second column contains the reconstructed and predicted frames respectively. Third column contains the respective heatmaps

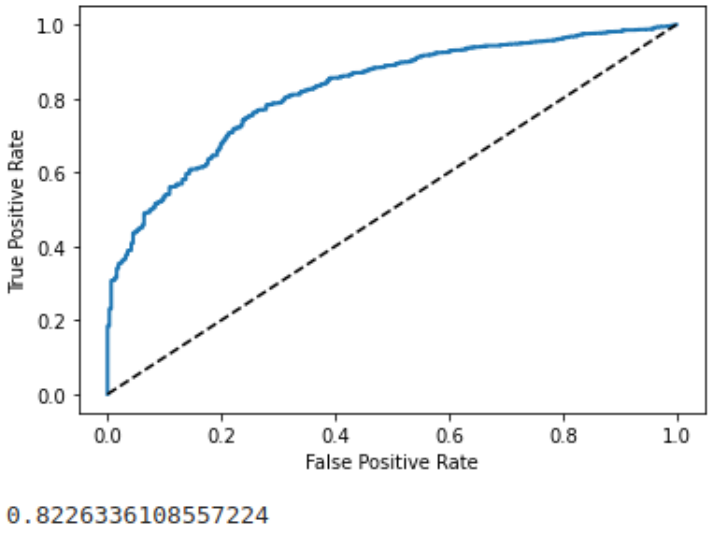

Figure 6.21: AUC-ROC curve for Ped2

6.4 Overall Results

After analysing the results, its obvious that skip connections leads to improvement in UCSD Ped2, but not in UCSD Ped1. Compared to the base model, there have been an increase of around 0.05 in AUC score for Ped2 in the final model. Moreover, the heatmaps for the base model with skip connections and data augmentation shows reduced number of high error pixels compared to that in base model and base model with skip connections. Qualitatively,iIt is evident from the Figure 6.6, 6.13, and 6.20. Pre-training on some larger video datasets might have increased the overall scores.

Overall scores for all the models are given in Table 6.7. S.C. and D.A. in Table 6.7 denotes Skip connections and Data Augmentation respectively.

Model	Ped1 AUC	Ped1 EER	Ped2 AUC	Ped2 EER
Base Model	0.768	0.258	0.776	0.305
Base Model with S.C.	0.764	0.294	0.796	0.274
Base Model with S.C. and D.A.	0.768	0.283	0.823	0.250

Table 6.7: Score for all the models

Chapter 7

Conclusion and Future Work

The model works fairly well for videos containing anomalous objects having relatively larger size, but it fails to identify the smaller anomalous objects. Moreover, after experimenting with the weights assigned to the reconstruction and prediction losses in the objective function, its evident that the spatial features plays a major role in detecting anomalies compared to the temporal features. Even in case of temporal anomalies, model seems to work well when the frames are sparsely crowded, but in case of normal but densely crowded frames, the model fails to generalize well leading to false alarms.

Future work involves further improving the model for smaller anomalous objects, as well as designing generalized models which might be able to perform fairly on all kinds of datasets. Moreover, designing better techniques to capture temporal features can be one of the research problem in this domain. Also, designing the models which can work fairly well, even in case of densely crowded images can be considered as good area for future research.

Most of the researchers tend to use small datasets, on which the scores have been saturated. New datasets with more challenging anomalies can be designed to make more generalized models.

Bibliography

[1] https://www.kaggle.com/code/shivamb/3d-convolutions-understanding-use-case

[2] http://www.svcl.ucsd.edu/projects/anomaly/dataset.html

[3] Chalapathy, R., Chawla, S.: Deep learning for anomaly detection: A survey (2019), https://arxiv.org/abs/1901.03407

[4] Cong, Y., Yuan, J., Liu, J.: Sparse reconstruction cost for abnormal event detection. In: CVPR 2011. pp. 3449–3456 (2011)

[5] Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences (2016), https://arxiv.org/abs/1604.04574

[6] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015), https://arxiv.org/abs/1502.03167

[7] Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1), 221–231 (2013)

[8] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classification with convolutional neural networks. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1725–1732 (2014)

[9] Kiran, B.R., Thomas, D.M., Parakkal, R.: An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos (2018), https://arxiv.org/abs/1801.03149

[10] Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: in ICML Workshop on Deep Learning for Audio, Speech and Language Processing (2013)

[11] Ramachandra, B., Jones, M.J., Vatsavai, R.R.: A survey of single-scene video anomaly detection. CoRR abs/2004.05993 (2020), https://arxiv.org/abs/2004.05993

[12] Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015), https://arxiv.org/abs/1505.04597

[13] Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.k., Woo, W.c.: Convolutional lstm network: A machine learning approach for precipitation nowcasting (2015), https://arxiv.org/abs/1506.04214

[14] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks (2014), https://arxiv.org/abs/1412.0767

[15] Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks (2014), https://arxiv.org/abs/1412.0767

[16] Xu, D., Ricci, E., Yan, Y., Song, J., Sebe, N.: Learning deep representations of appearance and motion for anomalous event detection (2015), https://arxiv.org/abs/1510.01553

[17] Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 2528–2535 (2010)

[18] Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., Hua, X.S.: Spatio-temporal autoencoder for video anomaly detection. In: Proceedings of the 25th ACM International Conference on Multimedia. p. 1933–1941. MM '17, Association for Computing Machinery, New York, NY, USA (2017), https://doi.org/10.1145/3123266.3123451

[19] Zhu, S., Chen, C., Sultani, W.: Video anomaly detection for smart surveillance (2020), https://arxiv.org/abs/2004.00222