# ABNORMAL EVENT DETECTION IN SURVEILLANCE VIDEOS USING TWO-STREAM DECODER

*Herman Prawiro*[*]*, Jian-Wei Peng*[†]*, Tse-Yu Pan*[*]*, and Min-Chun Hu*[*]

[*]Dept. of Computer Science, National Tsing Hua University, Taiwan
[†]Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan
herman.prawiro@mx.nthu.edu.tw, andersonpeng190@mislab.csie.ncku.edu.tw,
typan@mx.nthu.edu.tw, anitahu@cs.nthu.edu.tw

## ABSTRACT

Abnormal event detection in surveillance videos refers to the identification of events that deviate from the normal pattern. An autoencoder can be used to learn the normal patterns from the videos, and its reconstruction errors can be used to detect the abnormalities. Surveillance videos consist of two components: dynamic objects and a static background. Because of the nature of the static background, we can assume that the source of abnormality is from the objects. In this work, we propose the use of a two-stream decoder model to tackle the abnormal event detection problem in surveillance videos. The two-stream decoder comprised a background stream that models the static background and a foreground stream that models the dynamic objects. We also utilized a two-stream encoder to learn from optical flow, which contains motion information, and skip connections used to improve the details in the output frames. Several experiments on publicly available datasets were used to validate the effectiveness of the proposed model.

***Index Terms***— Abnormal event detection, surveillance videos, autoencoder, two-stream decoder

## 1. INTRODUCTION

In recent years, surveillance cameras have become ubiquitous in public places. One critical task is to detect abnormal events based on what is captured by the cameras. This task requires intensive human attention. What makes this effort seem futile is because an abnormal event occurs very rarely. Therefore, there is a need to develop an automated system to detect abnormal events in order to reduce human labor.

To develop such a system based on a supervised classification method is extremely challenging because the abnormality itself is unbounded, and it is infeasible to collect data corresponding to all abnormal events. Meanwhile, acquiring videos of an ordinary event is much easier. Abnormality is defined as a pattern that does not conform to expected normal behavior [1]. Thus, by learning the normal patterns from videos containing ordinary events, we can detect abnormal events as events that deviate from these learned patterns.

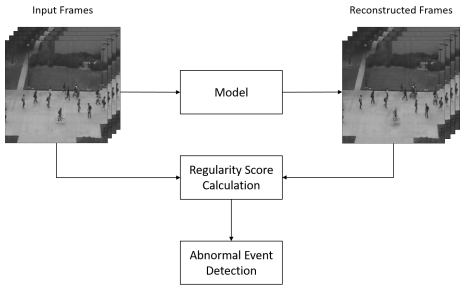


**Fig. 1**: Overview of abnormal event detection using autoencoder.

Fig. 1 shows an overview of abnormal event detection. Given that the only data available are normal videos, the task of abnormal event detection becomes an unsupervised learning problem. Hence, an autoencoder is naturally suited to this problem, as proposed by [2]. The autoencoder can learn the representation of normal videos by minimizing the reconstruction error of the input video. It is expected that if the autoencoder only learns the pattern of a normal scene, it will fail to reconstruct a scene with abnormalities. Therefore, we can use the reconstruction error as an indication of the presence of an abnormality. Formally, we normalize the reconstruction error into a regularity score, which has a value between 0 and 1, to decide whether the video contains abnormalities or not.

Typically, surveillance cameras are installed in places that overlook an area-of-interest and have a fixed angle and position. So, surveillance video is composed of dynamic objects and a static background. We can safely assume that the background is not the cause of an abnormality because it is static. Thus, the source of abnormality is from the objects. In order to detect an abnormality, it would be more beneficial to learn the objects' pattern.

Inspired by recent work from Vondrick *et al.* [3] to learn scene dynamics and to generate new video, we propose to use two-stream decoder to tackle the abnormal event detec-

978-1-7281-1485-9/20/$31.00 

tion problem. This decoder explicitly models the foreground separately from the background, in order to help the network focus on learning the appearance and motion of the objects. We then further explore this idea and add different components (e.g., skip connection and optical flow modality) to improve the model's performance in the abnormal event detection task.

The contributions of this work are summarized as follows. (1) We propose an approach for abnormal event detection in surveillance videos using a two-stream decoder without supervision. (2) We propose the utilization of a two-stream encoder and skip connections to further improve the model's performance.

## 2. RELATED WORK

Early research on unsupervised abnormality event detection mainly utilized hand-crafted features, e.g., optical flow and dynamic textures. Mehran *et al.* [4] proposed a social force model using optical flow to capture the dynamics of crowd behavior and to identify abnormal behavior. Lu *et al.* [5] and Zhao *et al.* [6] proposed the use of sparse coding or dictionary learning to encoder the normal patterns in the video.

The advancement of deep learning has made it possible to extract more robust and meaningful representations by which to achieve better performance in abnormal event detection. Xu *et al.* [7] proposed to train three Stacked Denoising Autoencoders (SDAEs) to extract features and One-Class SVM (OC-SVM) for each autoencoder as the classifier. Ionescu *et al.* [8] proposed the use of an unmasking technique on top of deep features, which removed the need for a training set.

Another approach to eliminating the need to train an additional classifier on top of extracted features is the use of deep autoencoders. This way, the network can be trained end-to-end by minimizing the reconstruction error. Hasan *et al.* [2] proposed the use of a 2D fully convolutional autoencoder to capture regularities from multiple datasets. Liu *et al.* [9] trained a model to predict a future scene instead of reconstructing the current scene. Gong *et al.* [10] proposed a memory-augmented autoencoder (MemAE).

While the CNN is easily able to learn spatial features from images, Recurrent Neural Networks (RNNs) and their variant, Long Short-Term Memory (LSTM), can learn from sequential data. Luo *et al.* [11] proposed Temporally-coherent Sparse Coding (TSC) that combined a sparse coding approach with an RNN. Although it is strong at sequential data, LSTM is not suitable for learning directly from a spatial image. Thus, many researchers have proposed the use of Convolutional LSTM (ConvLSTM) to obtain the advantages of both CNN and LSTM. Luo *et al.* [12] proposed to use a Convolutional LSTM Autoencoder (ConvLSTM-AE). Yan *et al.* [13] used a ConvLSTM Variational Autoencoder (VAE) model to learn from both image frames and optical flow.

However, to our knowledge, there has been still no previous work that leverages the fact that the surveillance camera is static. In this work, we attempt to explore the abnormal event detection problem with the assumption that we can model an object separately from its static background.

## 3. PROPOSED MODEL

Fig. 2 shows the proposed model. The model is an autoencoder comprising a two-stream encoder and two-stream decoder. The encoder and decoder are connected via skip connections.

### 3.1. Two-stream Decoder

Inspired by [3], we employ a two-stream decoder in the proposed model that explicitly models a static background and dynamic foregrounds. This decoder architecture has two components: a 2D convolutional background decoder and a 3D convolutional foreground decoder. The decoder itself can be formulated as:

$$dec(h) = m(h) \odot f(h) + (1 - m(h)) \odot b(h). \quad (1)$$

Here, $\odot$ denotes the Hadamard product or element-wise multiplication, and $h$ is the hidden representation extracted from the encoder. The output of the decoder is a combination of a dynamic foreground $f(h)$ stream and a static background $b(h)$ stream that is governed by a spatio-temporal mask $0 \leq m(h) \leq 1$. The background stream produces a spatial image that is replicated over time, while the foreground stream produces a spatio-temporal cuboid. The mask can select either foreground or background for each pixel location and timestep.

We use an architecture similar to [3], which follows the generator from DCGAN [14]. However, instead of using a deconvolution (transposed convolution) operation to generate higher resolution feature maps, we use nearest-neighbor upsampling followed by a convolution operation to avoid checkerboard artifacts [15] in the output images. The convolution, batch normalization, and ReLU operations are performed after the upsampling operation.

The background decoder consists of a five-layer 2D convolution network. It outputs a single background image. Each layer consists of a background decoder block and a nearest-neighbor upsampling operation. Each block has 2D convolution, batch normalization, and an ReLU layer. Every convolution layer has kernel size of 3x3 and stride of 1. The last layer outputs three-channel RGB images (or one channel for a grayscale image) and uses Tanh activation.

The foreground decoder consists of a five-layer 3D convolution network. It outputs two spatio-temporal cuboids (a stack of foreground images and a stack of masks) by splitting into two branches in the last layer. The mask branch outputs only one channel and uses Sigmoid activation. The

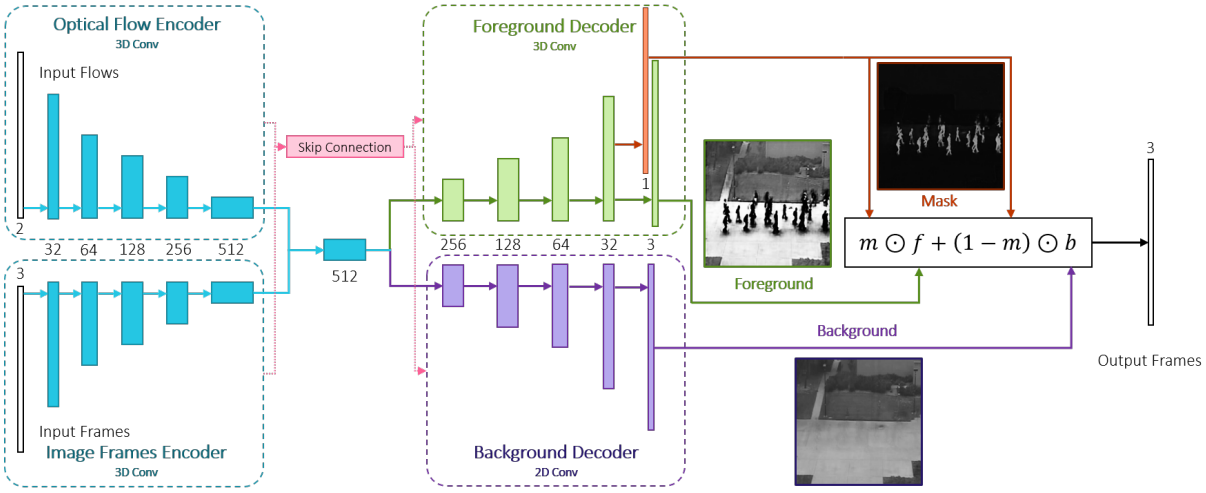


**Fig. 2**: The proposed autoencoder model with a two-stream encoder and a two-stream decoder. The encoder and decoder are interconnected with skip connections.

foreground branch outputs three-channel RGB images (or one channel for grayscale images) and uses Tanh activation.

### 3.2. Two-stream Encoder

The encoder mirrors the foreground decoder architecture and replaces the nearest-neighbor upsampling operation with the strided convolution (to downsample instead of to upsample). Every layer downsamples the spatial resolution by half and doubles the number of channels. Each layer is composed of a 3D convolution, batch normalization, and ReLU activation.

Simonyan *et al.* [16] proposed a two-stream convolutional network architecture, consisting of spatial and temporal stream for action recognition task. The spatial stream uses raw image frames as the input, whereas the temporal stream uses optical flow. Their work proved that the optical flow, which carries information about motion, is beneficial with regard to action recognition task. Driven by this insight, we utilize a two-stream encoder in the proposed model to help the network learn both the appearance and motion patterns in the image frame.

In the proposed two-stream encoder architecture, another encoder is added to learn from optical flow modality. Both encoders share the same architecture, but not weight. The optical flow from 2 subsequent frames is extracted using the TV-L1 algorithm. Then, the hidden representation from both encoders is fused into a single hidden representation by concatenating and using a convolution layer, as proposed in [17]. It allows our model to learn the correspondences between the two hidden representations and to reduce the dimensionality. The fusion layer consists of a 3D convolution with a kernel size of 1x1x1, batch normalization, and ReLU.

### 3.3. Skip Connection

To improve the output details, symmetrical skip connections [18] are utilized between the encoder and the decoder. Four encoder layers are connected to the corresponding decoder layers with the same feature map size. We use 1x1x1 convolution as bottleneck in the skip connection to reduce the number of channels by half for each encoder stream. We then concatenate the feature map from the skip connection and from the previous decoder layer.

### 3.4. Future Prediction

Liu *et al.* [9] proposed to predict future scene instead of reconstruct current scene. We also explore this approach with our model. The intuition is to force the network to learn the appearance and the motion of the objects in order to successfully predict future scenes. This approach is done by comparing the output of the model to its expectation of a future scene. Given a video clip with $T$ consecutive frames $I_1, I_2, ..., I_T$ as input, we attempt to predict the next $T$ consecutive frames $I_{T+1}, I_{T+2}, ..., I_{2T}$ as output. The model's architecture doesn't require modification to do future prediction.

### 3.5. Implementation and Training

We trained the network by minimizing the $L_2$ distance between the output frames $\hat{I}$ and its ground truth frames $I$.

$$\mathcal{L}_p(\hat{I}, I) = \|I - \hat{I}\|_2^2 \tag{2}$$

Given the nature of the background is static, we can help the background decoder to learn from an estimation of the background image $\bar{b}$. We computed $\bar{b}$ by calculating the mean from every frame in the training set. Then, we constrained the

network by minimizing the $L_2$ distance between the output of background stream $b(h)$ with the estimated background.

$$\mathcal{L}_{bg}(b(h), \bar{b}) = \|\bar{b} - b(h)\|_2^2 \quad (3)$$

We constrained the mask by enforcing a sparsity penalty to the mask by applying an $L_1$ penalty on the mask $m(h)$ to encourage the network to utilize the background stream more and to use the foreground stream only on dynamic objects.

$$\mathcal{L}_m(m(h)) = \|m(h)\|_1 \quad (4)$$

Finally, we combined the reconstruction loss $\mathcal{L}_p$, the background constraint $\mathcal{L}_{bg}$, and the mask constraint $\mathcal{L}_m$ with different weights $\lambda$ into a single objective function:

$$\mathcal{L}_{dec} = \lambda_p \mathcal{L}_p + \lambda_{bg} \mathcal{L}_{bg} + \lambda_m \mathcal{L}_m. \quad (5)$$

Pathak *et al.* [19] showed that using an adversarial loss [20], in addition to reconstruction ($L_2$) loss, can help the network to produce better results. The reconstruction loss captures the overall structure of an image, while the adversarial loss can pick a particular mode from the distribution, producing sharper and plausible image. Driven by this insight, we include an adversarial loss in our model.

Generative Adversarial Networks (GANs) [20] consist of a discriminative network $\mathcal{D}$ and a generative network $\mathcal{G}$. Our autoencoder can be treated as $\mathcal{G}$, and for $\mathcal{D}$, we use PatchGAN discriminator [21]. We use Least Squares GAN [22] to improve training stability. It uses Mean Squared Error (MSE) instead of log loss from the original GAN loss as objective function. LSGAN introduces the following adversarial loss:

$$\mathcal{L}_{adv}^{\mathcal{D}}(\hat{I}, I) = \frac{1}{2} MSE(\mathcal{D}(I), 1) + \frac{1}{2} MSE(\mathcal{D}(\hat{I}), 0) \quad (6)$$

$$\mathcal{L}_{adv}^{\mathcal{G}}(\hat{I}) = MSE(\mathcal{D}(\hat{I}), 1). \quad (7)$$

For $\mathcal{G}$, we combined Eq. 5 and Eq. 7 with a weight $\lambda_{adv}$ into a joint objective function, defined as:

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L}_{dec} + \lambda_{adv} \mathcal{L}_{adv}^{\mathcal{G}}. \quad (8)$$

To train the network, we resized the resolution of the image frame to 256 x 256 and normalized the value of each pixel to the range [-1, 1]. We constructed the input clip by randomly sampled $T = 4$ consecutive frames from training video. We used an Adam optimizer with a learning rate of 0.0002 and momentum parameters $\beta_1$ = 0.5 and $\beta_2$ = 0.999 and trained the model using a batch size of 8. We empirically set $\lambda_p$ = 1.0, $\lambda_{bg}$ = 0.2, $\lambda_m$ = 0.01, and $\lambda_{adv}$ = 0.05.

### 3.6. Abnormal Event Detection

To detect the abnormality in a video, we extracted a clip from the video using a sliding window of $T = 4$ consecutive frames $I_t$ with a step size of 1 frame. Then we passed the clip to the model to obtain an output clip $\hat{I}$.

We followed the procedure from [9] using the Peak Signal-to-Noise Ratio (PSNR) to measure the image quality, because it is better than MSE in terms of image quality assessment, as shown by Mathieu *et al.* [23]. Different from MSE, a high PSNR value indicates a higher quality image, which means that the frame is more likely to be normal, and vice versa. PSNR can be calculated as:

$$PSNR(\hat{I}_t, I_t) = 10 \log_{10} \frac{\max_I^2}{MSE(\hat{I}_t, I_t)}, \quad (9)$$

where $\max_I$ is the maximum possible pixel value of the image.

After calculating the PSNR for every frame in a clip, we averaged them to obtain the PSNR of the clip. Using the sliding window to extract the clip, resulted in fewer clips than the number of frames in the video. To overcome this issue, we interpolated the clip PSNRs to the number of frames in the video by using bicubic interpolation. Then, we calculated the regularity score $s(t)$ by normalizing the PSNR values in each testing video to the range [0, 1] using the following equation:

$$s(t) = \frac{PSNR(\hat{I}_t, I_t) - \min_t PSNR(\hat{I}_t, I_t)}{\max_t PSNR(\hat{I}_t, I_t) - \min_t PSNR(\hat{I}_t, I_t)}. \quad (10)$$

Therefore, we were able to predict the normality of the frame in the testing video based on its regularity score, where a high regularity score indicated that the frame was more likely to be normal. Then, we could set an arbitrary threshold to separate the regularity score into two classes and classify whether the frame is abnormal or not.

## 4. EXPERIMENTAL RESULTS

### 4.1. Datasets and Evaluation Metric

We evaluated the performance of our model using two publicly available datasets for abnormal event detection. The UCSD Pedestrian Dataset [24] is divided into two subsets: UCSD Ped1 and UCSD Ped2. Ped1 contains 34 training videos and 36 testing videos. Ped2 contains 16 training videos and 12 testing videos. The CUHK Avenue Dataset [5] contains 16 training videos and 21 testing videos. The dataset contains some unusual events (e.g., a person running, a person throwing objects). Each dataset only contains normal videos in the training set and both normal and abnormal videos in the test set.

Abnormal event detection can be defined as a binary classification problem where the goal is to classify whether the video is either normal or abnormal. Popular evaluation metric is to calculate the Receiver Operation Characteristic (ROC) by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The performance of different classifiers can be compared by calculating the Area Under Curve (AUC) of the ROC. Following the work from [24], we evaluate the performance using the frame-level AUC.

**Table 1**: The frame-level AUC (%) of the proposed method for different ablations evaluated on the Avenue, Ped1, and Ped2 datasets.

| Num of Encoders | Num of Decoders | Skip Connection | Future Prediction | Constraint | Adversarial Training | Avenue | Ped1 | Ped2 |
|---|---|---|---|---|---|---|---|---|
| One | One | - | - | - | - | 84.5 | 76.2 | 86.5 |
| One | Two | - | - | - | - | 84.9 | 76.7 | 87.6 |
| One | Two | √ | - | - | - | 85.2 | 75.1 | 92.0 |
| Two | Two | - | - | - | - | 84.1 | 77.0 | 89.4 |
| Two | Two | √ | - | - | - | 85.7 | 79.1 | 95.2 |
| Two | Two | √ | √ | - | - | 85.8 | 84.0 | 95.6 |
| Two | Two | √ | √ | - | √ | 85.6 | 83.8 | 95.5 |
| Two | Two | √ | √ | Background | - | 85.4 | 83.5 | 95.7 |
| Two | Two | √ | √ | Background | √ | 85.6 | 83.7 | 95.4 |
| Two | Two | √ | √ | Mask | - | 85.6 | 83.7 | 95.0 |
| Two | Two | √ | √ | Mask | √ | 85.5 | 83.7 | 95.0 |
| Two | Two | √ | √ | Background+Mask | - | 85.8 | 83.8 | 95.7 |
| Two | Two | √ | √ | Background+Mask | √ | **86.4** | **84.2** | **96.1** |

### 4.2. Model Ablation

**Comparison with the one-stream decoder**. We compared the frame-level AUC for the proposed two-stream decoder and one-stream decoder as a baseline. The one-stream decoder model used the same architecture as that of the foreground branch of the foreground decoder. Both models used only a single encoder with raw image frames as input. The results are shown in Table 1. The performance of the two-stream decoder was better than that of one-stream decoder, thus validating our assumption as shown in Fig. 3. Using two-stream decoder model improves detail of normal objects, while still struggled to reconstruct abnormal objects.

Following [9], regularity score gap $\Delta_s$ were used to evaluate our model architectures. It is a difference between average regularity score of abnormal frames and normal frames. Larger $\Delta_s$ means better capability to distinguish abnormality from normal patterns. The results are shown in Table 2.

**Table 2**: The regularity score gap ($\Delta_s$) of different model architectures evaluated on the Avenue, Ped1, and Ped2 datasets.

| | Avenue | Ped1 | Ped2 |
|---|---|---|---|
| Baseline (One-stream Decoder) | 0.344 | 0.235 | 0.346 |
| Two-stream Decoder | **0.356** | **0.245** | **0.358** |

**Impact of the two-stream encoder and skip connection**. As shown in Table 1, the model's performance is improved by adding skip connection and temporal information through two-stream encoder. In Fig. 3, although the model can reconstruct the cyclist, it still cannot properly reconstruct the bicycle. This indicated that the model didn't just blindly copy what it seen from skip connection.

**Impact of future prediction**. The results in Table 1 shows a significant performance improvement in the Ped1 dataset and a slight improvement in other datasets when predicting the future. The network is forced to learn more about motion in order to predict the future. The model will be more robust to motion, in addition to appearance.

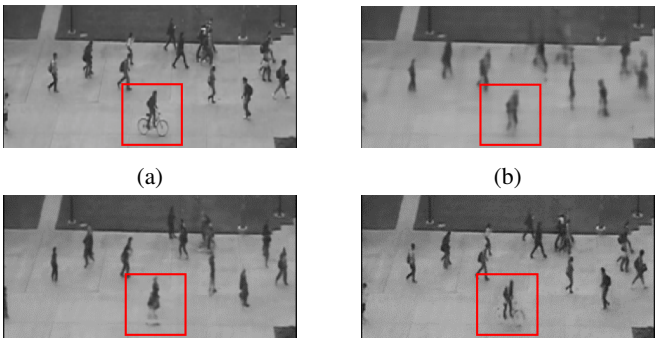


**Fig. 3**: A comparison of the output frame for different models. The red boxes denote an abnormality. (a) Ground truth. (b) One-stream decoder. (c) Two-stream decoder. (d) Two-stream decoder with two-stream encoder and skip connection.

**Impact of constraints and adversarial training**. From the result in Table 1, by combining the adversarial training with both constraints, our model achieves the highest AUC value. With both constraints and adversarial training, the model can better discriminate objects with normal motion and appearance, thus treating abnormal objects as background. Fig. 4 shows the model with both constraints and adversarial training has difficulties to reconstruct the abnormal objects.

### 4.3. Comparison with Existing Methods

In this section, we compared the performance of the proposed method with existing methods for abnormal event detection in surveillance videos. The results in Table 3 shows that our proposed method outperformed the existing methods in the CUHK Avenue and UCSD Pedestrian datasets.

(a) (b) (c)

**Fig. 4**: A comparison of an output frame for the proposed model with both background and mask constraints. The red boxes denote an abnormality. (a) Ground truth. (b) Without adversarial training. (c) With adversarial training.

**Table 3**: The frame-level AUC (%) of the proposed method and the other existing works evaluated on the Avenue, Ped1, and Ped2 datasets.

| Methods | Avenue | Ped1 | Ped2 |
|---|---|---|---|
| Conv-AE [2] | 70.2 | 81.0 | 90.0 |
| ConvLSTM-AE [12] | 77.0 | 75.5 | 88.1 |
| Unmasking [8] | 80.6 | 68.4 | 82.2 |
| Two-stream R-Conv-VAE [13] | 79.6 | 75.0 | 91.0 |
| Stacked RNN [11] | 81.7 | - | 92.2 |
| MemAE [10] | 83.3 | - | 94.1 |
| Future Frame Prediction [9] | 84.9 | 83.1 | 95.4 |
| **Ours** | **86.4** | **84.2** | **96.1** |

## 5. CONCLUSION

Since surveillance cameras have fixed angle and position, we proposed to use two-stream decoder to tackle abnormal event detection in surveillance videos. The two-stream decoder models both static background and dynamic objects. Our model utilized the two-stream encoder to learn from optical flow, which contains motion information, and skip connections to improve the detail of the output frames. Different constraints and adversarial training were applied to train a more robust model for abnormality event detection task. Our experiments showed that our method outperformed the existing methods for the abnormal event detection in the UCSD Pedestrian and CUHK Avenue dataset.

## 6. REFERENCES


[1] V. Chandola *et al.*, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, July 2009.

[2] M. Hasan *et al.*, "Learning temporal regularity in video sequences," in *Proceedings of CVPR*, 2016.

[3] C. Vondrick *et al.*, "Generating videos with scene dynamics," in *Proceedings of NIPS*, 2016.

[4] R. Mehran *et al.*, "Abnormal crowd behavior detection using social force model," in *Proceedings of CVPR*, 2009.

[5] C. Lu *et al.*, "Abnormal event detection at 150 fps in matlab," in *Proceedings of ICCV*, 2013.

[6] B. Zhao *et al.*, "Online detection of unusual events in videos via dynamic sparse coding," in *Proceedings of CVPR*, 2011.

[7] D. Xu *et al.*, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding*, vol. 156, 2017.

[8] R. T. Ionescu *et al.*, "Unmasking the abnormal events in video," in *Proceedings of ICCV*, 2017.

[9] W. Liu *et al.*, "Future frame prediction for anomaly detection - a new baseline," in *Proceedings of CVPR*, 2018.

[10] D. Gong *et al.*, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of ICCV*, 2019.

[11] W. Luo *et al.*, "A revisit of sparse coding based anomaly detection in stacked rnn framework," in *Proceedings of ICCV*, 2017.

[12] W. Luo *et al.*, "Remembering history with convolutional lstm for anomaly detection," in *Proceedings of ICME*, 2017.

[13] S. Yan *et al.*, "Abnormal event detection from videos using a two-stream recurrent variational autoencoder," *IEEE Transactions on Cognitive and Developmental Systems*, 2018.

[14] A. Radford *et al.*, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proceedings of ICLR*, 2016.

[15] A. Odena *et al.*, "Deconvolution and checkerboard artifacts," *Distill*, 2016.

[16] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proceedings of NIPS*, 2014.

[17] C. Feichtenhofer *et al.*, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of CVPR*, 2016.

[18] X.-J. Mao *et al.*, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proceedings of NIPS*, 2016.

[19] D. Pathak *et al.*, "Context encoders: Feature learning by inpainting," in *Proceedings of CVPR*, 2016.

[20] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proceedings of NIPS*, 2014.

[21] P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proceedings of CVPR*, 2017.

[22] X. Mao *et al.*, "Least squares generative adversarial networks," in *Proceedings of ICCV*, 2017.

[23] M. Mathieu *et al.*, "Deep multi-scale video prediction beyond mean square error," in *Proceedings of ICLR*, 2016.

[24] V. Mahadevan *et al.*, "Anomaly detection in crowded scenes," in *Proceedings of CVPR*, 2010.