# Abnormal Event Detection from Videos using a Two-stream Recurrent Variational Autoencoder

Shiyang Yan, Jeremy S. Smith, *Member, IEEE,* Wenjin Lu and Bailing Zhang

***Abstract*—With the widespread deployment of video surveillance systems the automatic detection of abnormal events in video streams has become increasingly important. An abnormal event can be considered as a deviation from the regular scene; however, the distribution of normal and abnormal events is severely imbalanced, since the abnormal events do not frequently occur. To make use of a large number of video surveillance videos of regular scenes, we propose a semi-supervised learning scheme, which only uses the data that contains the ordinary scenes. The proposed model has a two-stream structure that is composed of appearance and motion stream. For each stream, a recurrent variational autoencoder can model the probabilistic distribution of the normal data in a semi-supervised learning schemes. The appearance and motion features from the two streams can provide complementary information to describe this probabilistic distribution. Comprehensive experiments validate the effectiveness of our proposed scheme on several public benchmark datasets including Avenue, Ped1, Ped2, Subway-entry, and Subway-exit.**

***Index Terms*—Abnormal Event Detection, Variational Autoencoder, Convolutional LSTM, Reconstruction Error Probability, Two-stream Fusion**

## I. Introduction

The widespread deployment of surveillance cameras in airports, malls, and streets has resulted in the rapid increase of video data. A large workforce is often needed to process this video surveillance data due to the lack of computer vision solutions. To ensure the safety and security of the public environment, abnormal events, such as people fighting or urgent events like fire, should be detected quickly and accurately. However, abnormal events have a low probability of occurring, which makes manual detection a very tedious job. As a result, the automatic detection of rare or unusual incidents and activities in a surveillance video is urgently needed.

Generally, it is difficult to define an anomaly without a specific context. For example, running is a normal event on a football pitch but an abnormal event in other locations such as a restaurant. Hence, it is quite difficult to build a supervised learning model to discriminate these anomalies from normalities since only a small proportion count for the abnormal events. This is the well-known imbalance problem in machine learning [1]. Despite the efforts that equate anomaly detection with a binary classifier (normal and abnormal) [2], the scheme is often unrealistic, in real-world applications since the abnormal event footage in video sequences are rare, which makes the training of conventional classifiers impractical. As an alternative, recent research tries to accomplish abnormal event detection in a semi-supervised way, which only analyzes the distribution of ordinary data, and signifies the abnormal score during testing. Examples of this kind of scheme include the exploration of spatial-temporal features [3], dictionary learning [4], sparse representation [5] [6] and autoencoders [7] [8].

In the last two years, deep learning has become one of the most promising approaches for image processing, due to its excellent performance in various vision tasks including image classification [9] [10], object detection [11] and action recognition [12]. Deep neural networks can learn essential and discriminative features using their multi-layer non-linear transformations. It is therefore natural to apply deep neural networks to abnormal event detection in videos. Previous endeavours include the autoencoder-based approaches [7]. To detect an abnormal event in a video, an autoencoder tries to reconstruct the video frames and generates the reconstruction error which is considered as a regularity score. This can be considered as a kind of semi-supervised learning schemes in which an autoencoder is trained on the normal data to model its probability distribution through reconstruction. When testing, if there is an abnormal event in a video, the corresponding reconstruction error score is higher than the normal data since the model has not met the abnormal pattern during training. Hence, the comprehensive modeling of the normal data is of vital importance.

An inherited deficiency of the conventional autoencoder is its deterministic nature, which means no probabilistic interpretation or inference could be made about the data. Recently, a new generative model, called the variational autoencoder (VAE), has been proved to be a powerful tool [13]. A VAE with an autoencoder-like architecture is a directed probabilistic graphical model in which the posterior probability distributions are approximated by a neural network. Compared with a conventional autoencoder, VAE is unique as it encodes the original image into a prior distribution instead of deterministic features. Consequently, the VAE has shown superior results on some learning tasks such as image reconstruction and generation [13] [14]. Based on these considerations, we apply the VAE to abnormal event detection in videos.

Nevertheless, there are some obstacles to directly applying the vanilla VAE as it is targeted at static image reconstruction. How to capture the spatial-temporal features of a given video sequence is a primary question. It is well-known that the recurrent connections in a neural network is a powerful and effective way to model the dynamics of a sequence [15].

Shiyang Yan, Wenjin Lu and Bailing Zhang are with the Department of Computer Science and Software Engineering, Xi'an Jiaotong-liverpool University, Suzhou, China.

Jeremy S. Smith is with the University of Liverpool.

Manuscript received; revised

Hence, we apply recurrent units in the VAE model to better capture the temporal dependencies of the video frames. The widely known Long-Short Term Memory (LSTM) network [16] is the first option due to its advantages among the different recurrent neural networks (RNNs), particularly the solution to the gradient vanishing problem [17]. In this paper, we apply LSTM to learn the long-term dependencies of a video sequence. However, LSTM is known to be limited in its expression of spatial information, which is vital to the high-level visual semantics. To tackle the issue, we apply the convolutional LSTM [18] in which all the state-to-state transitions of memory cells are convolutional operations. The model can not only capture the temporal dependencies but also preserve the spatial information.

Recently, another developing approach for video processing in the deep learning framework is two-stream networks, which had been successfully applied in video-based action recognition [12] [19], often with state-of-the-art results. The method is also end-to-end learning, which is vital for many real-world applications. Two-stream networks extract features from both the spatial and temporal streams and then fuse them appropriately for subsequent processing [19]. The spatial stream is implemented with a CNN specializing in static frames [12] while the temporal stream uses another CNN, which is designed to extract temporal features. Compared with a general RNN scheme whose emphasis is on the temporal sequence modeling, the two-stream networks focus on the extraction of discriminative and complementary features. It is intuitive to combine the ideas from these two methodologies, which conforms to the proven effective practice of a combination of classifiers for different tasks [20]. For instance, [21] used two separate LSTM networks on the spatial stream and optical flows for action recognition. [22] systematically evaluated two-stream architectures for action recognition. In their research, the final performance is increased by employing LSTMs on both of the two streams. Optical flow can be considered as a low-level motion feature whilst LSTMs capture the long-term dependencies on the spatial features or the motion features. Hence, employing LSTMs and using two-stream fusion at the same time is advantageous; on the other hand, the long-term dependencies might be neglected when solely relying on the CNN-based model. For abnormal event detection, the information fusion from two streams is also expected to improve the system performance. In this paper, we set up a two-stream architecture for our VAE model with a novel double fusion scheme by utilizing both early fusion and late fusion.

In short, our contributions can be summarized as follows:

- We propose a novel model, namely the two-stream recurrent VAE, which provides a semi-supervised solution for abnormal event detection in videos.
- The recurrent VAE can model the probability distribution of video sequences by capturing the spatial-temporal features, and a two-stream architecture can learn features from both the spatial frames and optical flows. Subsequently, the advantage of the combination of the recurrent VAE and a two-stream architecture is validated.
- Our methods achieve improved results on the frame-level, event-level and pixel-level evaluations compared with current leading methods on several publicly available datasets.

## II. Related Works

### A. Abnormal Event Detection

As it is easier to obtain surveillance video data where the scene is normal, most research focused on the setting where the training data contains only normal visual patterns. Most video-based abnormal event detection approaches involve a local feature extraction step followed by learning a model using the training data, which only contains normal events. Any event that is an outlier from the learnt model is regarded as the anomaly [7]. This can be considered as a type of semi-supervised learning.

One of the popular local features is the trajectory-based feature. Trajectories have been very powerful in video processing and abnormal event detection [23] [24] [25] [26]. For example, Zhou et al. [26] proposed an abnormal event detection scheme based on the trajectory features and a Multi-Observation Hidden Markov Model (MOHMM) to detect abnormal events. Despite the fact that trajectory-based approaches have achieved successes in various video tasks [27] [28] [26], the dependence on tracking poses a bottleneck as it is still a challenge in computer vision. On the other hand, tracking-based methods are often not practical for crowded scenes event detection. Other local features include spatial-temporal features such as the histogram of oriented gradients [29] and the histogram of oriented flows [30].

Typical models based on these features in abnormal event detection include Bag of Visual Words (BoVW), where the local features are clustered in groups, according to some similarity metrics [31]. Sparse reconstruction is a similar codebook-based model in abnormal event detection [5]. For instance, [32] proposed detecting abnormal events via sparse reconstruction over the normal bases (dictionary). One of the advantages of sparse reconstruction is the suitability on modeling the high-dimensional data using relatively few training samples [32] [33]. Normal events are likely to generate sparse reconstructions with a small reconstruction cost while abnormal data tends to generate dense representation since the data is dissimilar with the pattern of normal data. Yu et al. [34] proposed to use Multi-scale Histogram of Optical Flow (MHOF) and Multi-scale Histogram of Gradient (MHOG) for feature representation and sparse models to detect abnormal events. Most of the codebook-based approaches, however, have the disadvantage of ignoring the spatial relationships among the image patches, which substantially limit their expression capability. On the other hand, the determination of the codebook size is often ad-hoc, which cannot guarantee optimal performance in real applications.

Some probabilistic graphical models have also been applied to abnormal event detection, e.g., the Hidden Markov Model (HMM) [31]. Similarly, the Conditional Random Field (CRF) can be used as the model to guarantee the global consistency of the anomaly judgments. For example, Li et al. [35] used a set of dynamic texture models to calculate the spatial and temporal abnormality maps, which are considered as the potential functions of the CRF model. Additionally, one-class Support Vector Machines (SVM) can be used to model the distribution of the normal patterns given the feature representations of the samples. For instance, M.Erfani et al. [36] studied large-scale anomaly detection using Deep Belief Networks (DBN) as the feature extractor and a one-class SVM to model the distribution of the normal data. These methods often consider feature extraction and classification as two separate components. During implementation, a memory device with a large capacity is also needed to store the high-dimensional feature vectors. It is also less practical since using one-class SVMs for this application typically needs at least two steps (the feature extraction and the classification) to finish the task.

Another effective and widely-used approach is based on autoencoders [37] [7]. An autoencoder [38] is a kind of neural networks that can be used for dimension reduction and image reconstruction. The successes of deep neural networks in various vision tasks consequently inspired autoencoder-based approaches in many vision tasks, including abnormal event detection. Neural networks based on deep learning architectures can automatically abstract different levels of features from the raw data. Researchers using the hierarchical structure of deep learning neural networks, e.g., deep convolutional neural networks (CNNs) have achieved great success in many tasks such as image classification [10], object detection [11], semantic segmentation [39] and action recognition [12]. Xu et al. [37] proposed a deep model for abnormal event detection which uses an autoencoder for feature learning and a linear classifier for abnormal event detection. Hasan et al. [7] proposed an end-to-end learning model using a stacked autoencoder for abnormal event detection in videos, with good results. To better capture the temporal dependencies of video frames, [8] proposed to use a LSTM embedded into the autoencoder, also with improved results. A similar idea of using the LSTM to capture temporal information has also been reported in [40] for time series anomaly detection. Sabokrou et al. [41] proposed to use an auto-encoder to learn features and a Gaussian classifier to distinguish the normal and the abnormal events in a semi-supervised learning scheme.

### B. Variational Autoencoder

As has been discussed previously, the conventional autoencoder is deterministic, which lacks the capability to interpret probabilistically or infer from the data. [42] applied a VAE [13] for anomaly detection from images using reconstruction probability. [43] proposed to combine the RNNs and the variational inference for anomaly detection in the time series data of a robot. Both of these two pieces of research demonstrated that the VAE-based models are better than the deterministic approaches, which inspired us to apply a recurrent VAE for abnormal event detection in video.

A VAE is an unsupervised learning approach for complicated distributions modeling [44]. It is a generative model parameterized by neural networks, which can be trained by the backpropagation algorithm.

Recently, the VAE has shown superior performance in several image processing tasks, e.g., image generation. [13] [45] applied VAEs to generate handwritten digits. [13] [46] [47] proposed generating images of faces using VAEs. [48] used the VAE to forecast future frames based only on static images. Moreover, the VAE can also be applied in a semi-supervised learning scheme. For instance, [49] extended the VAE to semi-supervised learning with class labels. Some traditional computer vision tasks such as image segmentation can benefit from VAE, for example, [50] proposed the use of a VAE to generate the segmentation map of an image.

Some hybrid learning systems have been proposed by the combination of VAE and other deep neural network models. [14] proposed an architecture incorporating convolutional neural networks into a VAE for image and caption generation. In their research, the deep generative deconvolutional network is used as a decoder of the latent variables whilst the convolutional neural network is used as the encoder of the given image. Also, the recurrent connection has been proposed to integrate into the VAE model to deal with sequence modeling [51]. The variational recurrent autoencoder [51] can be applied for efficient, large-scale unsupervised learning on time series data by mapping the time series data to a latent vector representation. [52] explored the inclusion of latent random variables into RNNs by combining the elements of the VAE, which can also be considered as one kind of recurrent VAE. Since we are dealing with video sequences which contain both the spatial and temporal information, convolutional operations and recurrent connections are both needed. The convolutional LSTM [18] preserves the convolution operation, which meets our requirement.

Meanwhile, the two-stream fusion method for action recognition in videos has achieved great success since the first publication [12]. Much subsequent research borrowed the idea from [12] in dealing with various vision problems [19] [53] [54]. Hence, we set up a two-stream recurrent VAE model, which is applied for semi-supervised learning of the data. Even though the two-stream idea had been widely applied, to the best knowledge of our knowledge, we are the first to propose a two-stream architecture for a VAE model.

## III. METHODOLOGY

### A. Variational Autoencoder

The VAE [13] is a recently proposed generative learning model [13]. A VAE introduces a set of latent random variables $z$, which are used to capture the variations in the input variables $x$. As one kind of directed graphical model, the joint distribution is defined in Equation 1.

$$p(x, z) = p(x|z)p(z) \tag{1}$$

The prior of the latent variables, $p(z)$, is generally chosen as a simple Gaussian distribution. The conditional probability $p(x|z)$ is parameterized by a highly flexible function approximator such as neural networks. This highly nonlinear mapping from $x$ to $z$ results in an intractable inference of the posterior $p(z|x)$. Hence, the VAE chose to use another distribution

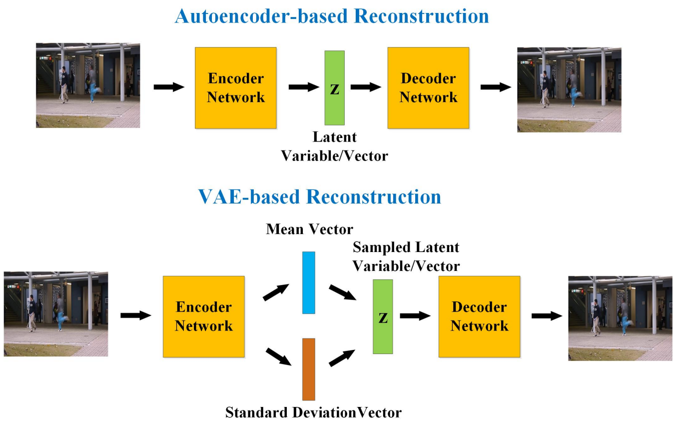



Fig. 1. Illustration of the VAE-based and autoencoder-based reconstruction schemes.

$q(z|x)$ as the posterior that enables the use of variational lower bound as explained in Equation 2.

$$logp(x) \geq -KL(q(z|x) \| p(z)) + E_{q(z|x)}(logp(x|z)) \quad (2)$$

where $KL(P\|Q)$ is the Kullback-Leibler divergence between two distributions $P$ and $Q$.

In VAE, the approximate posterior $q(z|x)$ is a Gaussian distribution whose mean $\mu$ and variance $\sigma^2$ are the outputs of the non-linear mapping, i.e., neural networks, from inputs $x$. Ideally, we would like to sample from this distribution. However, the stochastic gradient descent via back propagation cannot handle stochastic units inside a neural network. The solution for VAE is called the reparameterization trick, which is to move the sampling to an input layer. Given $\mu$ and $\sigma^2$, the mean and variance, we can firstly sample from a standard Gaussian distribution $\epsilon \sim N(0, I)$, then calculate $z = \mu + \sigma \cdot \epsilon$, where $\cdot$ indicates elementwise multiplication. The generative model $p(x|z)$ and inference model $q(z|x)$ are jointly trained by maximizing the variational lower bound.

A VAE-based image reconstruction scheme and a comparison with autoencoder-based reconstruction is shown in Fig. 1.

### B. The Convolutional LSTM

Our proposed recurrent convolutional VAE applies the convolutional LSTM as the basic building block for recurrent connections inside the VAE model. Hence, we firstly introduce the basic principle of the convolutional LSTM proposed in [18].

Let $\sigma(x) = (1+e^{-x})^{-1}$ be the sigmoid non-linear activation function and $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ be the tangent non-linear activation function, the convolutional LSTM model follows the following updating rules:

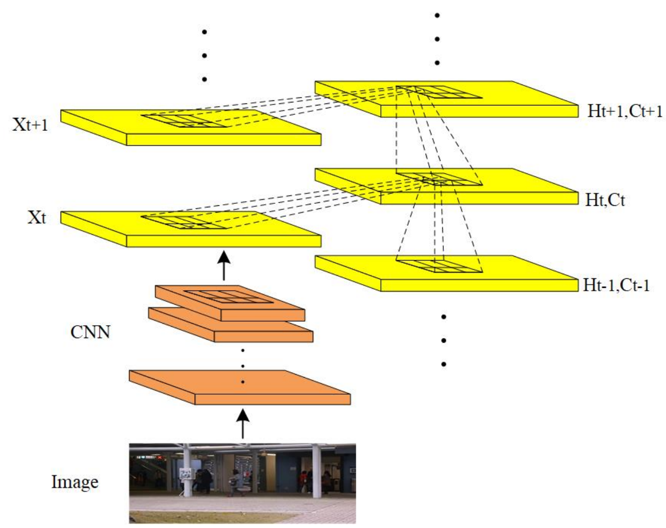



Fig. 2. The system diagram of convolutional LSTM

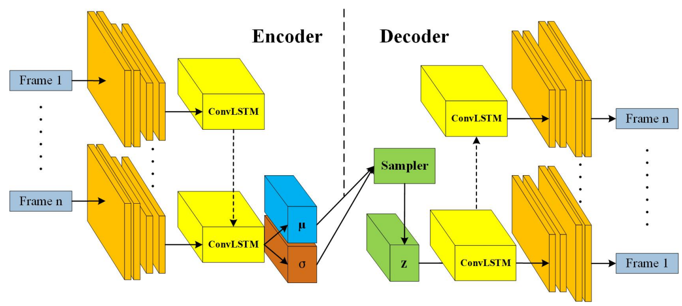



Fig. 3. The proposed recurrent convolutional VAE model

$$\begin{aligned}
i_t &= \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + b_o) \\
g_t &= \sigma(W_{xg} * x_t + W_{hg} * h_{t-1} + b_g) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot g_t \\
h_t &= o_t \cdot \phi(c_t) \\
y_t &= \phi(W_{yt} * h_t + b_y)
\end{aligned} \quad (3)$$

where $t$ is the time step in RNNs, $i_t, f_t, o_t$ are the input, forget and output gates of the LSTM model, respectively. $c_t$ is the cell memory while $h_t$ is the hidden state of the LSTM model. $g_t$ controls the update of the cell memory. $y_t$ is the output of the LSTM model. A $*$ indicates the convolution operation. $W_{\sim}, b_{\sim}$ are convolutional weights and bias, respectively. $x_t$ is the input to the LSTM model at each time step. Fig. 2 shows the system diagram of the convolutional LSTM.

### C. The Proposed R-ConvVAE Model

Blending LSTMs with the VAE architecture has been proposed previously to solve the natural language processing problem in [55], where a language generation model with both a LSTM and VAE is applied to explicitly model the holistic properties of sentences such as style, topic, and high-level syntactic features. [56] also introduced an LSTM-VAE neural network for the task of automated FAQs. These two pieces of research share the similar idea of combining an LSTM and a VAE to model the sequence. In this paper, a convolutional LSTM is embedded into the VAE to model the video sequence for abnormal event detection. We call this model R-ConvVAE.

In the proposed encoder network, video frames are firstly processed by a set of convolutional layers, followed by a convolutional LSTM block. The convolutional LSTM is set to capture the temporal dependencies of the video sequences. The distribution over the latent variable $z$ is obtained from the last state vector of the convolutional LSTM, which is described by Equation 4.

$$\begin{aligned} \mu &= W_\mu * h_{end} + b_\mu \\ log(\sigma) &= W_\sigma * h_{end} + b_\sigma \end{aligned} \tag{4}$$

where $h_{end}$ is the last hidden state of the convolutional LSTM. $\mu$ and $\sigma$ are the mean and variance of the latent variables. $W_\sim$ and $b_\sim$ are the convolutional weights and bias, respectively. $z$ can be obtained using Equation 5.

$$\begin{aligned} \epsilon &=\sim N(0, I) \\ z &= \mu + \sigma \cdot \epsilon \end{aligned} \tag{5}$$

where $\sim$ indicates the sampling operation.

Using the reparameterization trick, $z$ is sampled from the encoder. Then $z$ is used to initialize the hidden state of the convolutional LSTM of the decoder, which is followed by a set of deconvolutional operations for the reconstruction of each video frame. The proposed model is shown in Fig. 3.

### D. VAE for Abnormal Event Detection

We propose an abnormal event detection method which uses a recurrent convolutional VAE to calculate the anomaly score from the reconstruction error probability. The variational lower bound in Equation 2 is considered as the reconstruction error probability and reflects the probability distribution of the reconstruction of the original image.

The reconstruction error probability is different from the reconstruction error defined in conventional autoencoder-based abnormal event detection. Firstly, the latent variables $z$ in a VAE model are stochastic variables. However, in a conventional autoencoder, the hidden state $h$ is a deterministic variable. Also, the VAE model takes account of the variability of the latent variables by the procedure of sampling. This mostly extends the expressive power of the VAE model.

Also, reconstruction by a VAE is a stochastic process, which not only considers the difference between the reconstruction and the original data but also the variability of the distribution itself. This characteristic enables the VAE model to have a strong modeling capability for data, thus ensuring the generalization capability. This feature is missed in the conventional autoencoder, which makes its generalization capability poor.

In practice, we compute the reconstruction error probability of a pixel's intensity value $I$ at location $(x, y)$ in frame $t$ of a given video. From each frame, we compute the reconstruction error probability by summing up all the pixel-based probabilities. If we can define the reconstruction error probability of a frame as $p(t)$, the regularity score can be defined as in Equation 6.

$$s(t) = 1 - \frac{p(t) - min_t p(t)}{max_t p(t) - min_t p(t)} \tag{6}$$

The regularity score corresponds to the level of normality of each frame in the video. Like many detection scenarios such as object detection [57] [58], the regularity score plays a role in indicating the confidence of detection results. A preferable way to evaluate the detection performance with these confidence scores is to use the Receiver Operating Characteristic (ROC) curve, which will be further discussed in Section IV-C3.

### E. Two-stream Architecture for Abnormal Event Detection

We set up a two-stream architecture for abnormal event detection. The two-stream model for action recognition was proposed in [12] who proved that the temporal features of optical flow and deep spatial features are complementary. Different from the recognition tasks in [12], our motivation is to fuse the reconstruction error probabilities for abnormal event detection. Our idea of employing the two-stream architecture is that the temporal regularity of the appearance features and the motion features can be complementary in deciding the abnormal events in a semi-supervised scheme. Since the normal pattern needs to be modeled properly in the semi-supervised scheme, using a two-stream architecture is more comprehensive than a single stream.

The system model can be seen in Fig. 4. The spatial stream is to reconstruct the spatial frames using a recurrent VAE while the temporal stream is to reconstruct the stacked optical flows with a similar VAE network. Specifically, we use the GPU implementation of optical flow of [19], with a stride of 2. As the optical flow only captures the neighboring motion, it is desirable that LSTMs can be employed to capture the long-term dependencies. To model the probabilistic distribution of the motion features, we stack the vertical and horizontal parts of the optical flows into a two-channel image so as to compute the reconstruction error probabilities. The networks for both of the spatial stream and temporal stream are the recurrent convolutional VAE described previously, with a similar network structure.

To establish an early fusion scheme, we stack the static frame (gray image) and optical flow image (two-channel image) into a three-channel input to a recurrent convolutional VAE with the same architecture, which we call an early fusion stream. Once formulated as one image in the early fusion stream, the convolutional operation considers the static frame and optical flow image as a whole to compute the hierarchical features, level by level.

The spatial stream and temporal stream can be trained jointly and independently. During testing, the reconstruction error probabilities from the spatial stream and the temporal stream are fused by summation, which is denoted as late fusion. The difference between early fusion and late fusion was discussed in [19], which reveals that late fusion yields better performance.

The late fusion results can be added to the early fusion results for the reconstruction error probability fusion, denoted as double fusion. In this paper, we prove that the early fusion and late fusion provide complementary information. The fused reconstruction error probabilities are then utilized for post-processing and evaluation. Note that the post-processing corresponds to different operations in frame-level and event-level evaluations which are described in Section IV-C3.

## IV. Experiments

In this section, we introduce two use cases, the frame-level and pixel-level experimental procedures are used for different purposes: the frame-level detection is to find temporal regularity whilst the pixel-level detection is to localize the abnormal events in a video frame.

### A. Model Configuration

Table I shows the detailed configuration of the proposed model for the spatial stream. Specifically, the model contains an encoder and a decoder. The encoder consists of four convolutional layers, followed by a convolutional LSTM layer to capture the temporal information of the video frames. The decoder firstly uses a convolutional LSTM to decode the latent variable $z$ sampled from the encoder, followed by four deconvolutional layers to reconstruct the video frame.

A convolutional layer can connect multiple input activations within a fixed receptive field to a single activation output. On the other hand, a deconvolutional layer is to densify the sparse inputs by convolution-like operations with multiple filters. Hence, the spatial size of the output feature maps of a deconvolutional layer is larger than the spatial size of its corresponding inputs.

Also, there are two pooling layers after the Conv2 and Conv4 layers in the encoder network, and two unpooling layers after Deconv1 and Deconv4 in the decoder network. The max pooling operation in the encoder provides translation invariance. The unpooling layer in the decoder is to perform the reverse operation of pooling and reconstruct the original size of activations [59] [60] [61].

Table II presents the parameters of the architecture of the proposed model for the temporal stream. As we stack the vertical and horizontal parts of the optical flows, the inputs to the network are of depth 2. The other parameters are the same as that in the spatial stream.

### B. Datasets

We conducted experiments on several challenging datasets to test our methods. There are several public benchmark datasets targeting at abnormal event detection, namely, Avenue [3], UCSD pedestrian [62] and Subway datasets [63].

For the Avenue dataset, there are a total of 16 training and 21 testing video sequences. Each of the sequences is short, about 1 to 2 minutes long. The total number of training frames is 15,328 and there are 15,324 testing frames. The resolution of each frame is $640 \times 360$ pixels.

The UCSD pedestrian dataset contains two parts: UCSD-Ped1 and UCSD-Ped2. In UCSD-Ped1, there are 34 short clips for training, and another 36 clips for testing. All testing video clips have frame-level ground-truth labels, which indicate which frames of the video clip are abnormal. Each clip has 200 frames, with a resolution of $238 \times 158$ pixels. The UCSD-Ped2 has 16 short clips for training, and another 12 clips for testing. Each clip has 150 to 200 frames, with a resolution of $360 \times 240$ pixels.

In the Subway dataset, the videos are taken from two surveillance cameras in a subway station. One monitors the exit and the other monitors the entrance [63]. In both videos, the resolution is $512 \times 384$ pixels. The Subway-entry video is 1 hour 36 minutes long with 144, 249 frames in total, and the Subway-exit video is 43 minutes long with 64, 901 frames in total. All the testing videos have frame-level ground-truth labels.

### C. Frame-level detection

*1) Data Augmentation:* For frame-level abnormal event detection, following [7], we apply a data augmentation scheme to prepare for training because the available data is still not sufficient for the proposed model. Firstly, we extracted each frame from the raw video data, then resized it to a resolution of $227 \times 227$ pixels. As a common normalization practice in training the deep learning model, we subtract the global mean value of the pixels from each of the video frames. After that, the video frames are converted to grey scale images to reduce their dimensionality. All these operations are conducted using Matlab. The input to the model is a sequence of frames with a length of 10. To increase the size of the training data, we skip different strides to obtain the following frame sequences. For example, the first stride-1 sequence is composed of frames $\{1, 2 ,3, 4, 5, 6, 7, 8, 9, 10\}$. The stride-2 sequence is made of frames $\{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$, and the stride-3 sequence would contain frames $\{1, 4, 7, 10, 13, 16, 19, 22, 25, 28\}$. These operations can not only increase the size of the training data but also enable the model to capture long-term dependencies with the increase of skipped strides. For the temporal stream, we first stack the vertical and horizontal parts into a two-channel static image. Then the same data augmentation techniques are applied.

*2) Training Details:* [7] trained their convolutional autoencoder on all of the datasets together instead of on each individual one. [7] had proved that training on all the datasets does not influence the generalization capability of the model. Also, since we are dealing with a semi-supervised learning scheme based on the generative model, we do not use a pre-trained CNN model as in many deep learning schemes. We need a comparatively larger dataset in order to avoid the overfitting problem. Hence, we train our two-stream model on all the datasets used in this paper: Avenue, Ped1, Ped2, Entry and Exit.

We use an Adam optimizer [64] and a learning rate of 0.01 to train our recurrent VAE model from a Xavier uniform random weights initialization [65]. The batch size is set as 32. Usually, we find that the model converges in several epochs. The model was built using the Keras platform [65]. Moreover, all the experiments were undertaken on a PC equipped with a NVIDIA TITAN X GPU and running the Ubuntu 14.04 operating system.

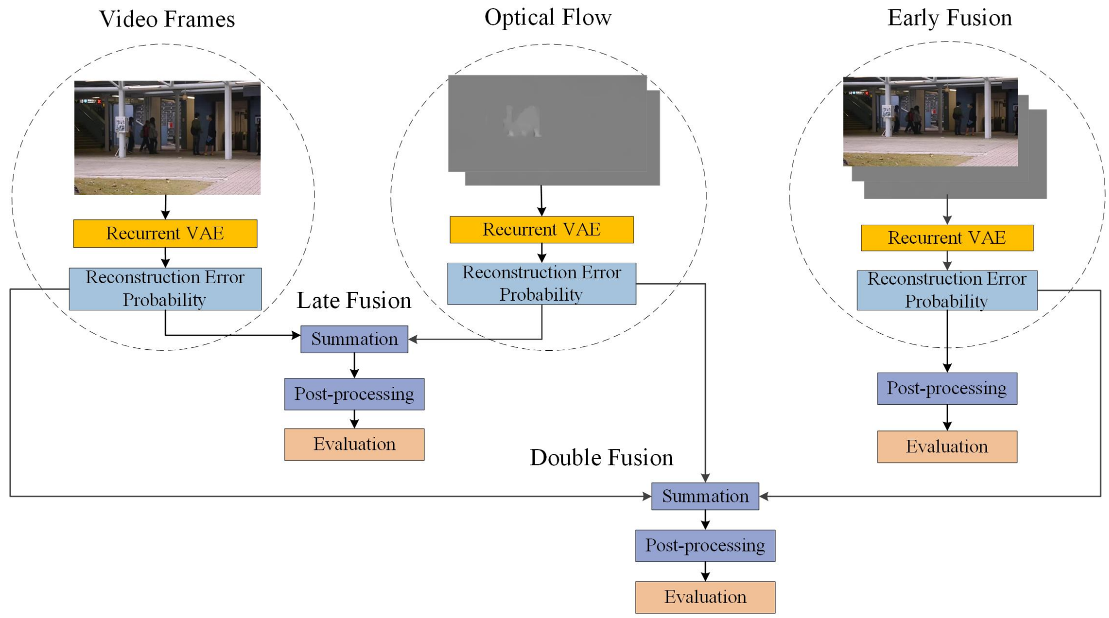



Fig. 4. Two-stream Architecture for Abnormal Event Detection

TABLE I
NETWORK CONFIGURATION FOR THE SPATIAL STREAM

| | | | | | | |
|---|---|---|---|---|---|---|
| Encoder | Image<br>1 × 227 × 227 | Conv1<br>128 × 55 × 55 | Conv2<br>128 × 27 × 27, Pooling | Conv3<br>64 × 27 × 27 | Conv4<br>64 × 13 × 13, Pooling | ConvLSTM<br>32 × 13 × 13 |
| Decoder | ConvLSTM<br>32 × 13 × 13 | Deconv1<br>64 × 13 × 13, Unpooling | Deconv2<br>64 × 27 × 27 | Deconv3<br>128 × 27 × 27 | Deconv4<br>128 × 55 × 55, Unpooling | Reconstruction<br>1 × 227 × 227 |

TABLE II
NETWORK CONFIGURATION FOR THE TEMPORAL STREAM

| | | | | | | |
|---|---|---|---|---|---|---|
| Encoder | Image<br>2 × 227 × 227 | Conv1<br>128 × 55 × 55 | Conv2<br>128 × 27 × 27, Pooling | Conv3<br>64 × 27 × 27 | Conv4<br>64 × 13 × 13, Pooling | ConvLSTM<br>32 × 13 × 13 |
| Decoder | ConvLSTM<br>32 × 13 × 13 | Deconv1<br>64 × 13 × 13, Unpooling | Deconv2<br>64 × 27 × 27 | Deconv3<br>128 × 27 × 27 | Deconv4<br>128 × 55 × 55, Unpooling | Reconstruction<br>2 × 227 × 227 |

*3) Evaluation Metrics for Frame-level Detection:* We evaluate the frame-level detection using two metrics, corresponding to the frame-level and event-level, respectively.

- Frame-level: If a frame contains at least one abnormal event, it is considered as a correct detection. These detections are compared to the frame-level ground-truth label. The Receiver Operating Characteristic (ROC) curve is used to measure the performance of the frame-level detection. To generate the ROC curve, the true positive rate (TPR) and the false positive rate (FPR) are calculated and plotted at various threshold settings of the confidence score of the detection outputs. The Area Under Curve (AUC) and the Equal Error Rate (EER) are the two metrics for evaluation based on the ROC curve [66].
- Event-level: This evaluation criterion was used in [7]. To reduce the noisy and meaningless local minima in the regularity score, they used the Persistence1D [67] algorithm to group local minima. In [7], they used a fixed temporal window of 50 frames to group local minima. In other words, local minima within 50 frames belong to the same abnormal event. We followed this practice to group the detected events and set the threshold as 0.2. The detected temporal windows which overlap by more than 50% with the ground-truth abnormal event windows are considered as a detection. Hence, this is an event-level evaluation criteria.

*4) Results:* To determine the best model for subsequent experiments, we first evaluated the frame-level detection in different control settings. The results of the control experiments on the spatial stream are shown in Table III. We tested the proposed recurrent variational autoencoder with different cost functions and the vanilla autoencoder with the same architecture. It can be seen, from the table, that the VAE-based model often yields better results than the conventional autoencoder and usually the Mean Squared Error (MSE) loss (which corresponds to the Euclidean Distance between the inputs and outputs) is better than the Binary Cross Entropy (BCE) loss for the AUC and EER results. This is because the MSE is a more straightforward indicater in the reconstruction tasks. [68] also reported the MSE loss generates a smaller reconstruction error than BCE for the stacked autoencoder. The results in Table III indicate that the proposed model with MSE yields the best performance. Hence, in the following experiments, we set the loss function as the MSE.

We also conducted experiments to validate the improvements brought by using the VAE and the convolutional LSTM. To be more specific, we re-implemented the ConvAE model described in [7] and followed the training procedure. We then implemented the ConvVAE model where we use the same structure of the ConvAE described in [7] but with a VAE training and inference algorithm. The results of the VAE model are shown in Table IV where is can be observed that our R-ConvVAE model generates the best results.

TABLE III
FRAME-LEVEL RESULTS OF DIFFERENT SETTINGS ON THE SPATIAL STREAM

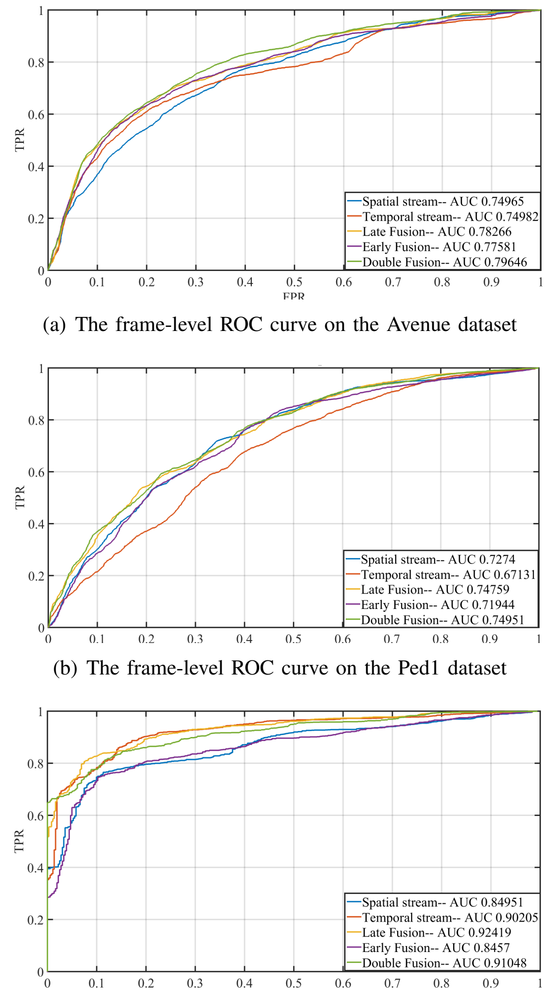
| Method | AUC/EER(%) | | | | |
|---|---|---|---|---|---|
| | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| R-ConvAE+BCE | 74.2/32.4 | 69.2/34.2 | 82.1/24.0 | 81.3/22.7 | 88.2/23.3 |
| R-ConvAE+MSE | 74.3/32.7 | 69.4/35.9 | 82.3/24.1 | 82.1/22.1 | 88.5/23.0 |
| R-ConvVAE+BCE | 74.8/**31.2** | 71.1/36.7 | 84.3/23.0 | 83.5/21.7 | 88.3/24.0 |
| R-ConvVAE+MSE | **75.0**/31.4 | **72.7/32.4** | **85.0/20.4** | **84.6/20.6** | **89.2/22.1** |

TABLE IV
SUMMARY OF FRAME-LEVEL RESULTS OF DIFFERENT APPROACHES ON THE SPATIAL STREAM

| Method | AUC/EER(%) | | | | |
|---|---|---|---|---|---|
| | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| ConvAE [7] (Our Results) | 73.5/31.6 | 72.6/33.0 | 83.3/23.1 | 84.2/21.6 | 88.1/24.4 |
| ConvVAE | 74.2/30.8 | 72.7/32.0 | 83.7/22.0 | 84.4/20.7 | 88.7/21.3 |
| R-ConvAE | 74.3/32.7 | 69.4/35.9 | 82.3/24.1 | 82.1/22.1 | 88.5/23.0 |
| R-ConvVAE | **75.0/31.4** | **72.7/32.4** | **85.0/20.4** | **84.6/20.6** | **89.2/22.1** |

Next, we tested the spatial and temporal streams for abnormal events detection using the proposed model. The results are shown in Table V. It is clear that using only the spatial or temporal stream by itself cannot generate the best result. However, with the information from the two-stream fused, the model has improved results compared with a single stream, which indicates that the information from the two-streams are complementary, and the two-streams fusion approach is an effective method. The early fusion described previously is not as good as late fusion. Nevertheless, our double fusion scheme can generate improved results, which can be seen in Table V. Fig. 5 and Fig. 6 show the ROC curves of the spatial stream, temporal stream, two-streams early fusion and two-streams late fusion on each of the five datasets.

In the proposed double fusion scheme, the spatial stream and temporal stream can be trained jointly, which means that the two streams share the latent prior probabilities but with different encoder and decoder networks. This structure can also be considered as a multi-task network in which one network performs two different tasks: the reconstruction of




(a) The frame-level ROC curve on the Avenue dataset

(b) The frame-level ROC curve on the Ped1 dataset

(c) The frame-level ROC curve on the Ped2 dataset

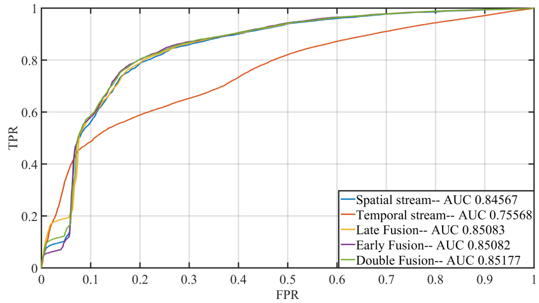
Fig. 5. ROC curve of the frame-level detection on the Avenue, Ped1 and Ped2 datasets

TABLE V
FRAME-LEVEL RESULTS OF THE TWO-STREAM FUSION

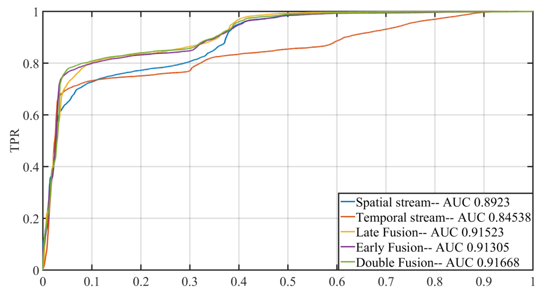
| Method | AUC/EER(%) | | | | |
|---|---|---|---|---|---|
| | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| Spatial Stream | 75.0/31.4 | 72.7/**32.4** | 85.0/20.4 | 84.6/20.6 | 89.2/22.1 |
| Temporal Stream | 75.0/30.4 | 67.1/37.2 | 88.3/18.2 | 75.6/33.0 | 84.5/24.1 |
| Two-Stream Early Fusion | 77.6/28.4 | 71.9/33.9 | 84.6/19.6 | 85.1/19.9 | 91.3/17.4 |
| Two-Stream Late Fusion | 78.3/28.1 | 74.8/32.7 | **92.4/15.2** | 85.1/20.4 | 91.5/17.0 |
| Two-Stream Double Fusion | **79.6/27.5** | **75.0/32.4** | 91.0/15.5 | **85.1/19.8** | **91.3/16.9** |

the static frames and the reconstruction of the optical flow images. Joint training performs slightly worse than independent training as shown in Table VI. One possible reason is that the prior distribution of the VAE can model more accurately when dealing with a single task. Hence, finally, we choose to train the spatial and temporal streams independently.

In Table VII, we compare our results with other published methods. The ConvAE proposed by Hasan, et al. [7], and R-ConvAE proposed in [8] are the closest results to ours. We achieve comparable results with these leading methods, and comparison experiments show that our methods improve on the baselines. A full list of the results can be seen in Table

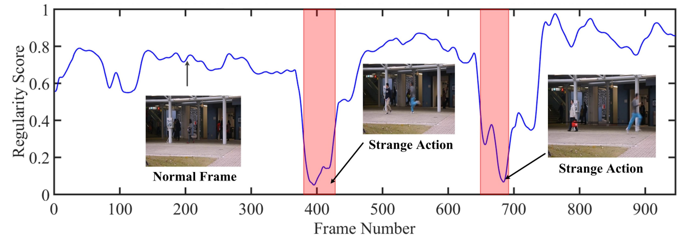



(a) The frame-level ROC curve on the Entry dataset

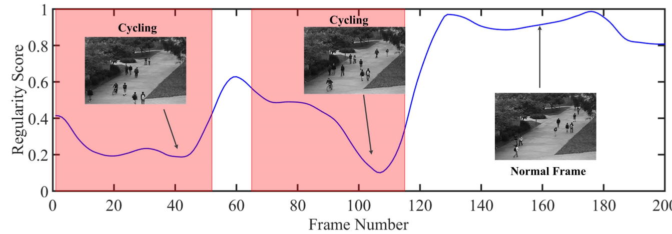



(b) The frame-level ROC curve on the Exit dataset

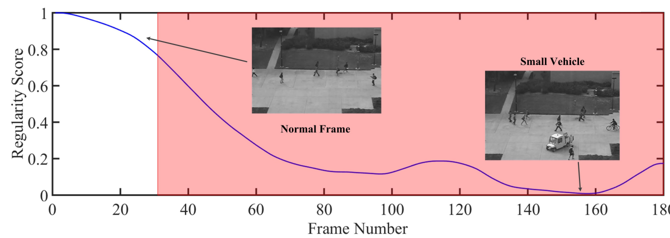
Fig. 6. ROC curve of the frame-level detection on the Entry and Exit datasets




(a) The visualization of abnormal events on video #4 of the Avenue dataset




(b) The visualization of abnormal events on video #32 of the Ped1 dataset




(c) The visualization of abnormal events on video #4 of the Ped2 dataset

Fig. 7. Visualization of the abnormal event detection on the Avenue, Ped1 and Ped2 datasets

TABLE VI
COMPARISON OF FRAME-LEVEL RESULTS OF TWO STREAM FUSION USING DIFFERENT TRAINING STRATEGIES.

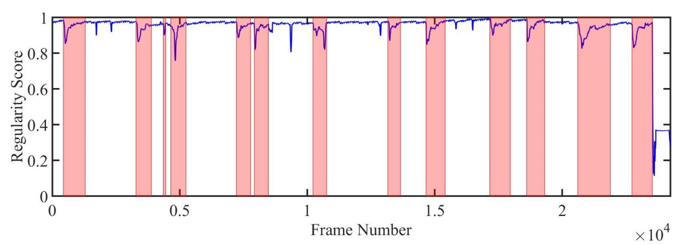
| Training Strategy | Method | AUC/EER(%) | | | | |
|---|---|---|---|---|---|---|
| | | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| Independent Training | Spatial Stream | 75.0/31.4 | 72.7/32.4 | 85.0/20.4 | 84.6/20.6 | 89.2/22.1 |
| | Temporal Stream | 75.0/30.4 | 67.1/37.2 | 88.3/18.2 | 75.6/33.0 | 84.5/24.1 |
| Joint Training | Spatial Stream | 71.1/35.2 | 70.0/33.4 | 84.2/21.2 | 84.4/20.7 | 89.7/21.3 |
| | Temporal Stream | 75.2/31.1 | 72.7/34.3 | 84.0/22.8 | 68.6/37.9 | 80.0/22.7 |

VII.

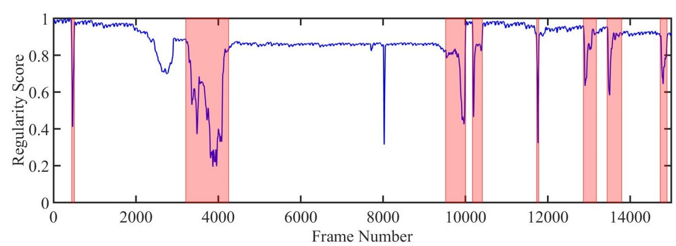
TABLE VII
FRAME-LEVEL RESULTS AND COMPARISON WITH OTHER METHODS

| Method | AUC/EER(%) | | | | |
|---|---|---|---|---|---|
| | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| Adam [63] | | 77.1/38.0 | -/42.0 | | |
| SF [69] | | 67.5/31.0 | 55.6/42.0 | | |
| MPPCA [62] | | 66.8/40.0 | 69.3/30.0 | | |
| MPPCA+SF [62] | | 74.2/32.0 | 61.3/36.0 | | |
| HOFME [70] | | 72.7/33.1 | 87.5/20.0 | 81.6/22.8 | 84.9/17.8 |
| ConvLSTM [71] | 84.0/- | 67.0/- | 77.0/- | - | - |
| ConvLSTM-AE [71] | 50.0/- | 43.0/- | 25.0/- | - | - |
| VAE [71] | 78.0/- | 63.0/- | 72.0/- | - | - |
| ConvAE [7] | 70.2/25.1 | **81.0**/27.9 | **90.0**/21.7 | **94.3**/26.0 | 80.7/9.9 |
| ConvAE [8] | 74.5/- | 68.1/- | 81.1/- | 91.0/- | 80.2/- |
| R-ConvAE [8] | **77.0**/- | 75.5/- | 88.1/- | 93.3/- | **87.7**/- |
| ConvAE (Our Results) | 73.5/31.6 | 72.6/33.0 | 83.3/23.1 | 84.2/21.6 | 88.1/24.4 |
| R-ConvAE (Our Results) | 74.3/32.7 | 69.4/35.9 | 82.3/24.1 | 82.1/22.1 | 88.5/23.0 |
| Two-Stream R-ConvVAE (Our Results) | **79.6/27.5** | **75.0/32.4** | **91.0/15.5** | **85.1/19.8** | **91.7/16.9** |

Following [7], we also evaluated the event-level detection on each of the five datasets. Table VIII shows the experimental results of event-level detection. From the table, the spatial stream tends to have better performance than the temporal stream. For instance, on the Avenue dataset, the spatial stream detects 36 abnormal events with 8 false alarms while the temporal stream detects 32 abnormal events with 12 false

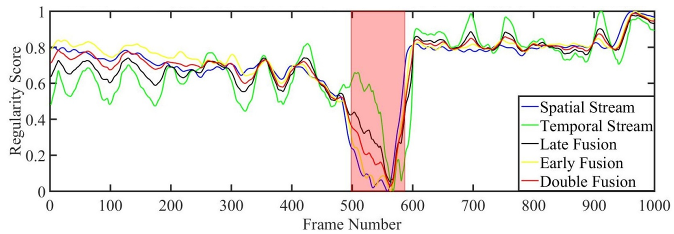



(a) The visualization of regularity scores on video #6 of the Entry dataset

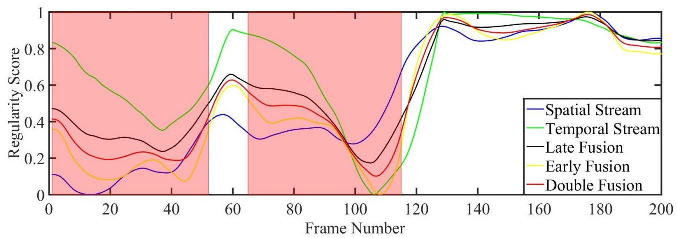



(b) The visualization of regularity scores on video #3 of the Exit dataset

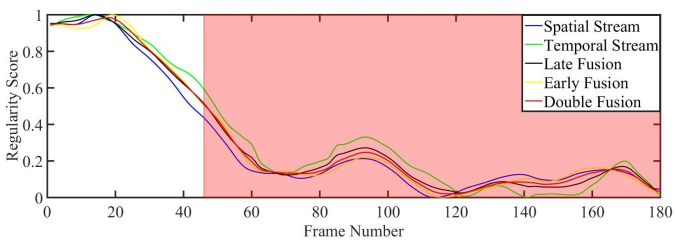
Fig. 8. Visualization of the regularity scores on the Entry and Exit datasets




(a) The visualization of abnormal events on video #15 of the Avenue dataset




(b) The visualization of abnormal events on video #32 of the Ped1 dataset




(c) The visualization of abnormal events on video #7 of the Ped2 dataset

Fig. 9. Visualization of the regularity scores of different streams and their fusion. In the figures, the red regions indicate the ground-truth frames of abnormal events.

alarms. On most of the datasets, our two-stream fusion method tends to have less false alarms when detecting abnormal events. We outperform the methods in [7] on the Ped1, Ped2 and Exit datasets.

TABLE VIII
EVENT-LEVEL RESULTS OF THE PROPOSED MODEL

| Method | Correct Detection/False Alarm | | | | |
|---|---|---|---|---|---|
| | Avenue | Ped1 | Ped2 | Subway Entry | Subway Exit |
| Abnormal Events | 47 | 40 | 12 | 66 | 19 |
| ConvAE [7] | 45/4 | 38/6 | 12/1 | 61/15 | 17/5 |
| Spatial Stream | 36/8 | 38/9 | 12/0 | 51/6 | 17/5 |
| Temporal Stream | 32/12 | 37/15 | 12/0 | 51/9 | 16/6 |
| Two-Stream Late Fusion | 32/6 | 37/5 | 12/0 | 52/8 | 18/4 |
| Two-Stream Early Fusion | 35/7 | 38/6 | 12/0 | 54/7 | 17/4 |
| Two-Stream Double Fusion | 34/6 | 38/5 | 12/0 | 56/7 | 18/4 |

*5) Discussion and Visualization:* Since the proposed scheme is a frame-level abnormal event detection method, the model does not locate the exact pixel position. During testing, the model generates a reconstruction error probability for each frame. The user only needs to analyse the frame-level reconstruction error probability to detect abnormal events. Regarding the system efficiency, our model takes approximately 0.0012s to generate a single reconstruction error probability on a Titan X (Maxwell Architecture) GPU. The testing time of the ConvAE, ConvVAE and R-ConvAE are same level, since they are all end-to-end learning models.

To better analyze the performance of our abnormal events detection scheme, we also plot the regularity score from each of the five datasets in Fig. 7 and Fig. 8. Fig. 7 provides the detected events and the corresponding regularity scores on the Avenue, Ped1, and Ped2 datasets. It is clear from the figure that the lower regularity scores correspond to abnormal events while high regularity scores correspond to normal frames. Fig. 8 provides the visualization of regularity scores for the Entry and Exit datasets, where the red color regions indicate the frame-level ground-truth label of abnormal events. As can be seen from the figure, the detection results match well with the ground-truth frames. We also compared the regularity score curve of the spatial, temporal, two-stream early fusion, two-stream late fusion and two-stream double fusion in Fig. 9. In the figure, the red curve indicates the regularity scores of the two-stream double fusion, which normally better correspond to the ground-truth abnormal frames.

### *D. Pixel-level Detection*

*1) Training and Testing Configurations:* The previously discussed training and testing scheme can be considered as a type of frame-level abnormal event detection method since this framework follows the research of [7], which is a typical deep learning temporal regularity detection scheme (frame-level). To enable the pixel-level abnormal event localization, a patch-based training and testing method is also carried out to test the feasibility of the proposed R-ConvVAE model.

Instead of training all the datasets together, in the patch-based training scheme, we train each dataset separately. Explicitly, we validate the R-ConvVAE model on two datasets, the Avenue, and Ped1 datasets, in which pixel level abnormal masks are provided for evaluation.

Firstly, following [2], a temporal-spatial foreground cube is detected. The frames in a video are first divided into some non-overlapping patches, using sliding windows. Then the foreground segmentation mask is generated by the Vibe algorithm [72]. Then the overlapping ratio between the foreground segmentation mask and each of the non-overlapping patches is computed: if the overlapping ratio is above 10 percent, the corresponding patch is recognized as a foreground. These patches, are then used to form the temporal-spatial cubes for a video: each patch that is considered to be foreground is used to form a cube with a sequence of 10 frames. After ignoring the duplicated cubes of a video, a set of temporal-spatial cubes is collected, and is ready for training. By doing so, the training efficiency is improved since only the foreground patches are used for training, the large portion of the video which contains only the background is ignored. The stride for the collection of cubes is set as 2 to guarantee there is enough data for training.

During testing, we also feed the video frames to the foreground detection algorithm to extract foreground patches to speed up the process and also filter out some of the false positive detections which might appear in the background regions. The whole video is segmented into several temporal-spatial cubes with 10 frames. For each cube in the video, we

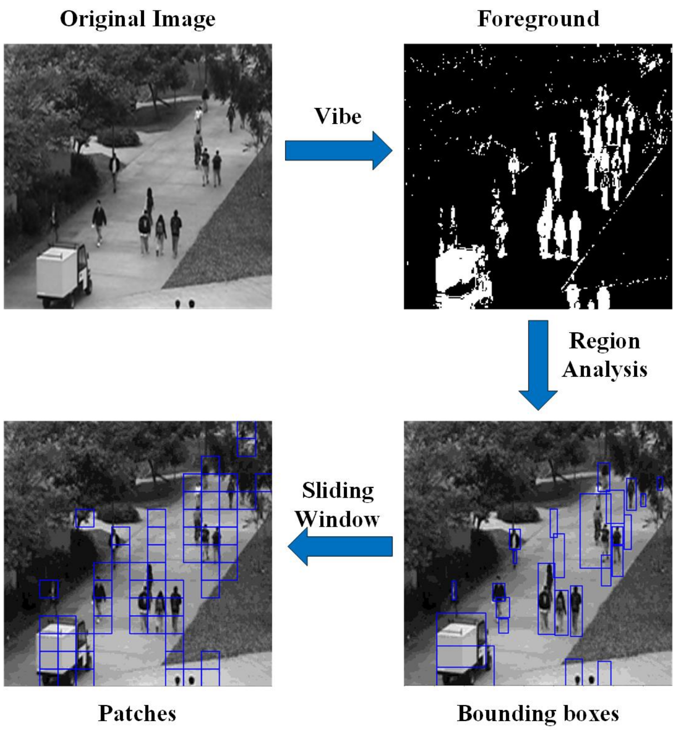



Fig. 10. Foreground detection for patch generation

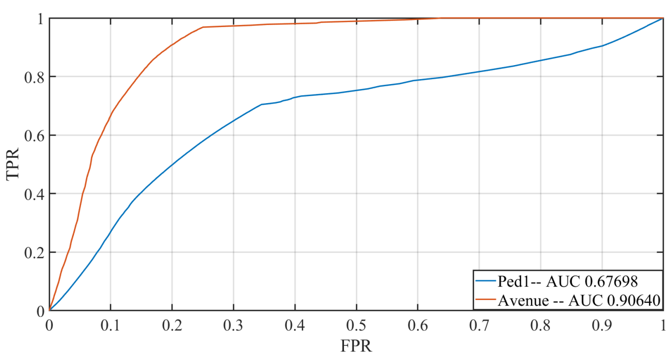



Fig. 11. Pixel-level ROC curves on the Avenue and Ped1 datasets

employ the same practice discussed previously to calculate the reconstruction error probability for each pixel. The pre-precessing steps can be seen in Fig. 10.

*2) Results and Visualization:* The ROC curves of the pixel-level evaluation of the Avenue and Ped1 dataset are shown in Fig. 11. The corresponding AUC value and comparison with previously-published results are presented in Table IX. As can be seen from the table, our R-ConvVAE method achieves the best AUC result on the Ped1 dataset, and with a satisfying result on the Avenue dataset. Our model only uses the appearance features to test the feasibility of the localization task. Also, the results using ConvAE, R-ConvAE, ConvVAE, and R-ConvVAE show consistency with the findings reported previously. A visualization of the detected abnormal regions on the Avenue dataset is shown in Fig. 12.

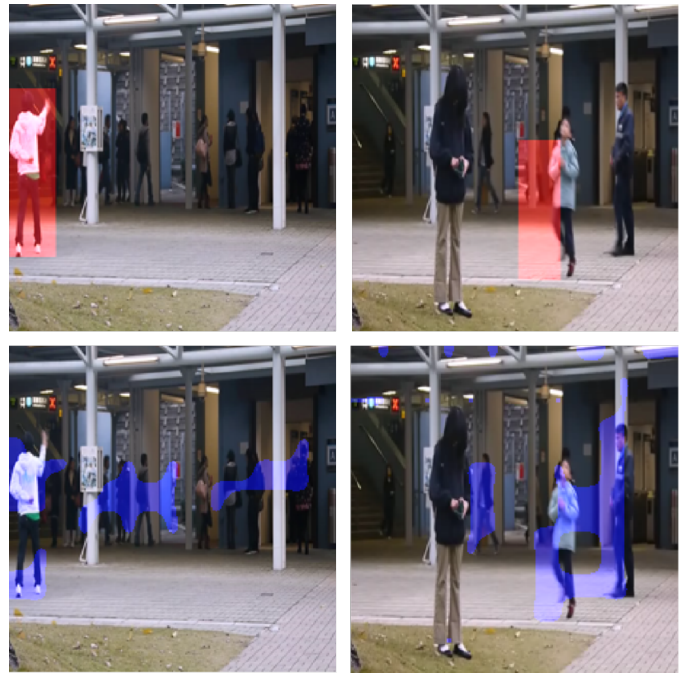


Fig. 12. Pixel-level detection results on the Avenue dataset: Top row are the ground-truth masks whilst the bottom row are the detection results. The frame is the 2nd frame of the 8th video from the Avenue dataset.

TABLE IX
PIXEL-LEVEL RESULTS AND COMPARISON WITH OTHER METHODS

| Method | AUC(%) | |
|---|---|---|
| | Avenue | Ped1 |
| Adam [63] | - | 46.1 |
| SF [69] | - | 19.7 |
| MPPCA [62] | - | 20.5 |
| MPPCA+SF [62] | - | 21.3 |
| Lu et al. [3] | **92.9** | 63.8 |
| Ren et al. [73] | - | 56.2 |
| Xu et al. [37] | - | 67.2 |
| Sum et al. [74] | - | 65.1 |
| Del Giorno et al. [75] | 91.0 | - |
| Zhang et al. [6] | - | 67.6 |
| Ours (ConvAE) | 89.1 | 65.5 |
| Ours (R-ConvAE) | 91.0 | 67.0 |
| Ours (ConvVAE) | 90.3 | 67.5 |
| Ours (R-ConvVAE) | 90.6 | **67.7** |

## V. CONCLUSION

To detect abnormal events from videos in a semi-supervised learning scheme, we proposed a two-stream recurrent VAE. The VAE is used to form a probability distribution of normal data by probability inference and reconstruction. The recurrent connection using a convolutional LSTM inside a VAE can preserve the spatial information whilst simultaneously capturing the long-term dependencies of video frames. The two-stream fusion architecture also demonstrates a powerful information fusion capability in abnormal event detection. The proposed model was tested on five publicly available datasets, namely Avenue, Ped1, Ped2, Subway-entry and Subway-exit, with improved results over other published methods.

## References


[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, Sept 2009.

[2] S. Zhou, W. Shen, D. Zeng, M. Fang, Y. Wei, and Z. Zhang, "Spatial–temporal convolutional neural networks for anomaly detection and localization in crowded scenes," *Signal Processing: Image Communication*, vol. 47, pp. 358–368, 2016.

[3] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2720–2727.

[4] S.-H. Yen and C.-H. Wang, "Abnormal event detection using hosf," in *IT Convergence and Security (ICITCS), 2013 International Conference on*. IEEE, 2013, pp. 1–4.

[5] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognition*, vol. 47, no. 5, pp. 1791–1799, 2014.

[6] Z. Zhang, X. Mei, and B. Xiao, "Abnormal event detection via compact low-rank sparse learning," *IEEE Intelligent Systems*, vol. 31, no. 2, pp. 29–36, 2016.

[7] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 733–742.

[8] W. Luo, W. Liu, and S. Gao, "Remembering history with convolutional lstm for anomaly detection," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*, July 2017, pp. 439–444.

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[13] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations (ICLR)*, 2014.

[14] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, "Variational autoencoder for deep learning of images, labels and captions," in *Advances in Neural Information Processing Systems*, 2016, pp. 2352–2360.

[15] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*, 2015, pp. 843–852.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[18] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.

[19] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[20] P. Ksieniewicz, B. Krawczyk, and M. Woźniak, "Ensemble of extreme learning machines with trained classifier combination and statistical features for hyperspectral data," *Neurocomputing*, 2017.

[21] Y. Wang, S. Wang, J. Tang, N. O'Hare, Y. Chang, and B. Li, "Hierarchical attention network for action recognition in videos," *arXiv preprint arXiv:1607.06416*, 2016.

[22] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao, "Towards good practices for very deep two-stream convnets," *arXiv preprint arXiv:1507.02159*, 2015.

[23] C. Li, Z. Han, Q. Ye, and J. Jiao, "Abnormal behavior detection via sparse reconstruction analysis of trajectory," in *Image and Graphics (ICIG), 2011 Sixth International Conference on*. IEEE, 2011, pp. 807–810.

[24] X. Mo, V. Monga, R. Bala, and Z. Fan, "Adaptive sparse representations for video anomaly detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 4, pp. 631–645, 2014.

[25] C. Piciarelli, C. Micheloni, and G. L. Foresti, "Trajectory-based anomalous event detection," *IEEE Transactions on Circuits and Systems for video Technology*, vol. 18, no. 11, pp. 1544–1554, 2008.

[26] S. Zhou, W. Shen, D. Zeng, and Z. Zhang, "Unusual event detection in crowded scenes by trajectory analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1300–1304.

[27] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3169–3176.

[28] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[30] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *European conference on computer vision*. Springer, 2006, pp. 428–441.

[31] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1446–1453.

[32] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3449–3456.

[33] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Robust visual tracking via structured multi-task sparse learning," *International journal of computer vision*, vol. 101, no. 2, pp. 367–383, 2013.

[34] Y. Yu, W. Shen, H. Huang, and Z. Zhang, "Abnormal event detection in crowded scenes using two sparse dictionaries with saliency," *Journal of Electronic Imaging*, vol. 26, no. 3, p. 033013, 2017.

[35] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 18–32, 2014.

[36] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[37] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Computer Vision and Image Understanding*, vol. 156, pp. 117 – 127, 2017.

[38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[39] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[40] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*. Presses universitaires de Louvain, 2015, p. 89.

[41] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 56–62.

[42] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," Technical Report, Tech. Rep., 2015.

[43] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational inference for on-line anomaly detection in high-dimensional time series," *International Conference on Learning Representations (ICLR) Workshops*, 2016.

[44] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[45] T. Salimans, D. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 1218–1226.

[46] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[47] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems*, 2015, pp. 2539–2547.

[48] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *European Conference on Computer Vision*. Springer, 2016, pp. 835–851.

[49] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[50] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.

[51] O. Fabius and J. R. van Amersfoort, "Variational recurrent auto-encoders," *International Conference on Learning Representations (ICLR) Workshops*, 2015.

[52] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, 2015, pp. 2980–2988.

[53] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1449–1457.

[54] A. Diba, A. M. Pazandeh, and L. Van Gool, "Efficient two-stream motion and appearance 3d cnns for video classification," in *European Conference on Computer Vision Workshops*, 2016.

[55] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 2016, pp. 10–21.

[56] M. Patidar, P. Agarwal, L. Vig, and G. Shro, "Correcting linguistic training bias in an faq-bot using lstm-vae," in *DMNLP Workshop of ECML-PKDD*, 2017.

[57] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 1440–1448.

[58] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, June 2017.

[59] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[60] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[61] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2018–2025.

[62] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1975–1981.

[63] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 3, pp. 555–560, 2008.

[64] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.

[65] F. Chollet *et al.*, "Keras," 2015.

[66] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[67] Y. Kozlov and T. Weinkauf, "Persistence1d: Extracting and filtering minima and maxima of 1d functions," *http://people. mpi-inf. mpg. de/ weinkauf/notes/persistence1d. html, accessed*, pp. 11–01, 2015.

[68] T. Amaral, L. M. Silva, L. A. Alexandre, C. Kandaswamy, J. M. Santos, and J. M. de Sá, "Using different cost functions to train stacked auto-encoders," in *Artificial Intelligence (MICAI), 2013 12th Mexican International Conference on*. IEEE, 2013, pp. 114–120.

[69] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 935–942.

[70] T. Wang and H. Snoussi, "Histograms of optical flow orientation for abnormal events detection," in *Performance Evaluation of Tracking and Surveillance (PETS), 2013 IEEE International Workshop on*. IEEE, 2013, pp. 45–52.

[71] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *arXiv preprint arXiv:1801.03149*, 2018.

[72] O. Barnich and M. V. Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, vol. 20, no. 6, pp. 1709–1724, June 2011.

[73] H. Ren, W. Liu, S. I. Olsen, S. Escalera, and T. B. Moeslund, "Unsupervised behavior-specific dictionary learning for abnormal event detection." in *BMVC*, 2015, pp. 28–1.

[74] Q. Sun, H. Liu, and T. Harada, "Online growing neural gas for anomaly detection in changing surveillance scenes," *Pattern Recognition*, vol. 64, pp. 187–201, 2017.

[75] A. Del Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 334–349.