REMEMBERING HISTORY WITH CONVOLUTIONAL LSTM FOR ANOMALY DETECTION

Weixin Luo[†]*, Wen Liu*[†]*, Shenghua Gao*

Shanghaitech University, Shanghai, China
{luowx, liuwen, gaoshh}@shanghaitech.edu.cn

ABSTRACT

This paper tackles anomaly detection in videos, which is an extremely challenging task because anomaly is unbounded. We approach this task by leveraging a Convolutional Neural Network (CNN or ConvNet) for appearance encoding for each frame, and leveraging a Convolutional Long Short Term Memory (ConvLSTM) for memorizing all past frames which corresponds to the motion information. Then we integrate ConvNet and ConvLSTM with Auto-Encoder, which is referred to as ConvLSTM-AE, to learn the regularity of appearance and motion for the ordinary moments. Compared with 3D Convolutional Auto-Encoder based anomaly detection, our main contribution lies in that we propose a ConvLSTM-AE framework which better encodes the change of appearance and motion for normal events, respectively. To evaluate our method, we first conduct experiments on a synthesized Moving-MNIST dataset under controlled settings, and results show that our method can easily identify the change of appearance and motion. Extensive experiments on real anomaly datasets further validate the effectiveness of our method for anomaly detection.

***Index Terms*—** Anomaly detection, Convolutional Neural Networks, Long Short Term Memory

1. INTRODUCTION

Anomaly detection is an important task in computer vision on account of its prevalent applications in video surveillance, video summarization, and scene understanding, *etc*. However, this task remains extremely challenging because it is an ill-posed problem, i.e., the scenarios of abnormal event are unbounded because it is extremely difficult or infeasible to collect data corresponding to all abnormal events. In contrast, the acquisition of ordinary moments in videos is much easier. Thus, a common setting for anomaly detection is that there are only ordinary moments available in the training sets. [1] Anomaly detection can be casted as the following two subproblems: i)how to characterize the appearance and motion; ii)how to model the change in appearance or motion. For quite a long time, hand-crafted features [1][2] are utilized to characterize the appearance and motion in videos, then sparse representation based approaches [3][4][5] can be used to measure the change of appearance or motion. However, such sparse representation strategy is very time-consuming for both training and testing. Recently, deep neural networks have shown their advantages over hand-crafted features for visual data representation in image classification [6] and activity recognition [7][8]. Recently, Hasan *et al.* [9] propose to use a 3D Convolutional Neural Network (ConvNet or CNN) based Auto-Encoder framework to simultaneously learn the regularity among the appearance and motion for anomaly detection. However, many existing work for activity recognition have shown that 3D convolution is not good enough for motion characterization [10][11].

In light of the success of CNN for image representation [6] and Long Short Term Memory (LSTM) for modeling the change of sequential data [7], in this paper, we propose to use ConvNet to encode each frame and use a Convolutional LSTM (ConvLSTM) [12], a variant of LSTM that preserves the spatial information, to memorize the change of the appearance which corresponds to motion information. Then we integrate CNN and ConvLSTM within an Auto-Encoder framework to guarantee ConvLSTM memorizes the past information. We use a Deconvolutional Network (DeconvNet) to reconstruct past frames, and to identify whether an anomaly occurs, we also reconstruct current frame with a different DeconvNet. Thus the reconstruction error is an indicator of the change in appearance or motion. We term our framework as ConvLSTM based Auto-Encoder (referred to as ConvLSTM-AE). As shown in experiments on synthesized Moving-MNIST dataset (Figure 2 and Table 1), compared with [9], our model can easily identify the change in appearance and motion, therefore our framework is more suitable for anomaly detection.

We summarize our contributions in this work as follow: i) we develop a ConvLSTM-AE framework to encode appearance and change of appearance (motion) for anomaly detection; ii) Experiments on a synthetic moving MNIST dataset show that our proposed ConvLSTM-AE can easily detect the anomaly caused by motion or appearance. Experiments on real datasets further verify the effectiveness of our framework for anomaly detection.

[1] †: Equal Contribution

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

2. RELATED WORK

Based on the representation for videos, one can roughly categorize anomaly detection methods into hand-crafted features based methods and deep neural networks based methods. All these approaches usually involve two steps, i.e., extract features for appearance or/and motion representation [1], and learn a model in an unsupervised way, such as sparse representation [4], Markov Random Field (MRF) [13], etc., to explore the regular patterns in the training data. We refer readers to [14] for a comprehensive review of hand-crafted features based methods. Recently, deep neural networks have shown their successes for many computer vision tasks [6][15], thus researchers have proposed to use deep neural networks for anomaly detection [9][16]. Because there is no abnormal events in training phase, usually an Auto-Encoder based approach is adopted. For regular appearance/motion, the reconstruction error of the learnt auto-encoder is low while for abnormal appearance or motion, the reconstruction error is high. Xu *et al.* [16] propose a multi-layer auto-encoder for representation learning. They first train three sub-networks for appearance, motion, and joint representation of appearance and motion. Then they train three one-class SVMs and fuse their output together for anomaly detection. Similarly, Hasan [9] proposes to use a 3D convolutional neural network to encode the appearance and motion information of a video clip. Then it use a deconvolutional neural network to reconstruct the input video clip. However, many existing work [10][11] show that 3D convolution cannot encode the motion very well. Further, [9] predicts the anomaly for all clips in a video. In order to get a frame level anomaly prediction, it has to do the anomaly detection for multiple video clips and interpolate the degree of anomaly for each frame, which is time-consuming.

3. APPROACH

For one thing, CNN has demonstrated its good performance for image representation [6] and motion information extraction for activity recognition [17]. For another thing, recurrent neural network (RNN) has demonstrated its speciality for encoding the motion in videos [18]. Thus we propose to combine both CNN and LSTM which is a special model of RNN, for regularity modeling in videos.

3.1. Convolutional LSTM

LSTM has shown its success for modeling the motion information for activity recognition [17]. Recently, a Convolutional LSTM (ConvLSTM) framework is proposed [12] which replaces the matrix multiplication with convolutional operation in LSTM. Compared with LSTM, ConvLSTM preserves the spatial information, therefore it facilitates the data reconstruction [12]. Specifically, we show the formulations of ConvLSTM in Equation (1), where x_t and h_t stand for the input and output of ConvLSTM at time t, and i_t, f_t and o_t stand for input, forget and output gates, respectively. A memory cell c_t stores the historical information. $\otimes$ represents the convolution operation, $\circ$ represents the element-wise multiplication and σ is the sigmoid activation function.

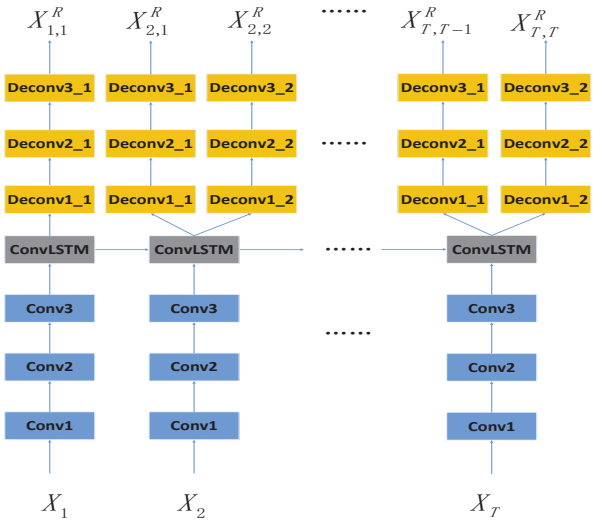


Fig. 1: The unfolded architecture of our ConvLSTM-AE framework. Conv modules denote convolution layers. Deconv modules denote deconvolution layers. ConvLSTM modules denote Convolution LSTM layer. All layers in the same row of the figure are identical. For each DeconvNet of each frame except for the first one, the left one reconstructs the previous frame and the right one reconstructs the current frame, and the DeconvNet corresponding to the first frame just reconstructs the first frame.

$$
\begin{aligned}
i_t &= \sigma(w_{xi} \otimes x_t + w_{hi} \otimes h_{t-1} + b_i) \\
f_t &= \sigma(w_{xf} \otimes x_t + w_{hf} \otimes h_{t-1} + b_i) \\
o_t &= \sigma(w_{xo} \otimes x_t + w_{ho} \otimes h_{t-1} + b_i) \\
c_t &= f_t \circ c_{t-1} + i_t \circ tanh(w_{xc} \otimes x_t + w_{hc} \otimes h_{t-1} + b_i) \\
h_t &= o_t \circ tanh(c_t)
\end{aligned}
\tag{1}
$$

3.2. Network Architecture

The network architecture of our ConvLSTM-AE is shown in Figure 1. In ConvLSTM-AE, a CNN is used to encode the content of each frame at first, then the encoded feature of each frame is fed into a ConvLSTM which memorizes the all historic frames. The output of ConvLSTM would be used to reconstruct the current frames and the previous frames. The intention of reconstructing previous frames is to guarantee that the ConvLSTM memorizes the information in historic frames. Now that the output of ConvLSTM contains the his-

toric information, for ordinary moments in the training data, it would help the reconstruction of current frames, thus the reconstruction error or current frames would be low. For abnormal events, with the change of appearance or motion, historic information cannot help the reconstruction of current frames. Then the reconstruction error will be high. Thus we can identify whether anomaly occurs or not based on reconstruction error.

Especially, the input of our network is T consecutive frames: $X_1, X_2, \ldots, X_T$. We denote the mapping function of the ConvNet in our ConvLSTM-AE as $x_t = f_c(X_t)$ for the t-th frame, denote the mapping function for the ConvLSTM-AE as $h_t = g(f_c(X_t), f_c(X_{t-1}), \ldots, f_c(X_1))$, and denote the reconstructed previous frames and current frames by the DeconvNets at time t as $X^R_{t,t-1} = f^p_d(h_t)$ and $X^R_{t,t} = f^c_d(h_t)$, respectively. For ordinary moments, it is desirable that the reconstructed frames should be close to the original inputs, thus we arrive at the following object function (When $t = 1$, we only reconstruct $X_{1,1}$.):

$$\arg\min \frac{1}{2} \sum_t ||X_t - X^R_{t,t}||^2_F + ||X_{t-1} - X^R_{t,t-1}||^2_F \quad (2)$$

For the ConvNets based encoding module, we firstly stack three convolution layers with 128, 256, 512 feature maps, respectively. The kernel sizes corresponding to these convolutional layers are 7×7, 5×5 and 3×3, respectively. For all convolutional layers, the stride is set to 2. Then we feed the output of ConvNet into a ConvLSTM-AE, and the output size of ConvLSTM-AE is the same with that of its input. In ConvLSTM-AE, h_0 and c_0 which correspond to the initial output and initial memory are set to 0. As for the DeconvNet based decoding module, we flip the architecture of ConvNet, i.e., the number of features corresponding the layers from bottom to up is 256, 128, and 1, and the kernel sizes are 7×7, 5×5 and 3×3, respectively, and for deconvolutional layers in DeconvNets, the stride is set to 2. Zero padding is used for all convolutional and deconvolutional layers.

3.3. Anomaly Detection with ConvLSTM

Once the model is trained, theoretically, our model can be used for anomaly detection on testing videos with arbitrary frames. However, our model is trained with video clips which contains T frames. Even in the testing phase, our model still memorizes the information in previous frames, which may correspond to anomaly events. Therefore, for better detection accuracy, each time we only do the anomaly prediction for a video clip containing T' consecutive frames. In other words, we enforce the network to forget all history information every T' frames to improve the anomaly detection accuracy. [2] Then we can calculate the reconstruction error corresponding to these T' frames. We repeat this for all clips from the whole testing video. When current input is I_t, the reconstruction error ($\ell(t)$) corresponding to current input can be calculated as follow:

$$\ell(I_t) = ||I_t - I^R_{t,t}||^2_F + ||I_{t-1} - I^R_{t,t-1}||^2_F \quad (3)$$

Following the work of [9], we normalize $\ell(I_t)$ based on the whole sequence of the testing video, i.e.,

$$s(t) = 1 - \frac{\ell(t) - \min_t \ell(t)}{\max_t \ell(t) - \min_t \ell(t)} \quad (4)$$

Here $s(t)$ can be used to determine when an abnormal event occurs. For an ordinary moment, $s(t)$ corresponds to a larger value. However, for abnormal event, $s(t)$ corresponds to a small value.

4. EXPERIMENTS

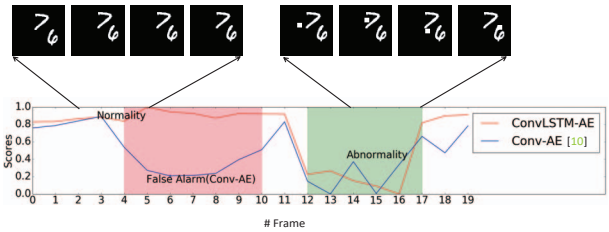
In this section, we empirically evaluate our proposed method on a synthesized data (the Moving-MNIST dataset [12]) and some publicly available anomaly detection datasets, including the CUHK Avenue dataset [4] ,the UCSD Pedestrian dataset [20], and the Subway Entrance and Exit dataset [21].

4.1. Implementation details

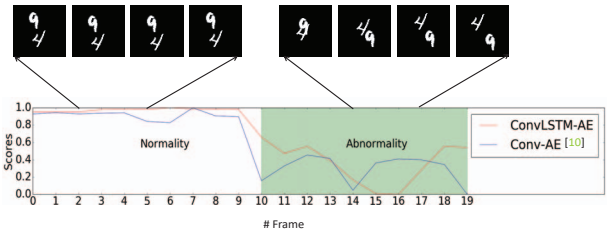
Our implementation is based on caffe framework[22]. To train our model more robustly, we use the same data augmentation strategy the same with [9], i.e., we set the stride between two sampled frames in original videos to 1, 2, and 3. The learning rate is set to 0.01, and AdaGrad [23] based stochastic gradient descent is used for the parameters optimization. The number of samples within each mini batch is 4. In our experiments, Xavier[24] initialization is used, which empirically corresponds to a higher accuracy. The activation functions in both ConvNet and DeconvNet are the rectified linear unit (ReLU). The resolution of each frame is resized to the size of 225 × 225.

As for the traing/testing split, we use the standard training/testing split for all real anomaly detection datasets. For evaluation metrics, following the work of [9], we compute a score for a frame and alter thresholds for each score of a frame. Then we can get the ROC curve and calculate the Area Under Curve (AUC). Higher AUC corresponding to better performance. Because of the limited space constraint, we compare our method with Conv-AE [9] and the work of [19] which are the latest work [9][19] and correspond to the state-of-the-art performance for anomaly detection. The results of Conv-AE [9] reported in the paper are based on the codes provided by the authors of the paper.

[2] Interestingly, in a recent work [19], the events frequently occurs wouldn't be regarded as anomaly. In this paper, we use the same anomaly definition as that in [9].



(a) Change in Appearance



(b) Change in Motion

Fig. 2: The score of some testing samples in Moving-MNIST dataset. The upper one denotes the change in appearance and the bottom one denotes the change in motion. The red curve represents ConvLSTM-AE and the blue one represents Conv-AE. The red region denotes the false alarm for Conv-AE while our ConvLSTM-AE predicts the labels of these frames correctly.

4.2. The Length of Video Clip for Training and Testing

We found that the effect of the length of T is insignificant for the performance, which agrees with the finding in [9]. And we set T to 10 for all datasets in our experiments, which is the same with [9]. As aforementioned, our network also memorizes historic frames in testing phase. Larger T' means more information are memorized. For the scenes with frequent changes, we can use a small T' to guarantee a higher accuracy. More specifically, T' is fixed to 10 for Avenue, Ped1 and Ped2 where people walk in/away frequently in these videos. And for Subway-entrance and Subway-exit, T' is set to the length of testing videos, which means we do not need to refresh the memory because in the Subway dataset, almost all frames correspond to the same background.

Table 1: AUC on Moving-MNIST

	Conv-AE[9]	ConvLSTM-AE
appearance	0.743	0.999
motion	0.940	0.949

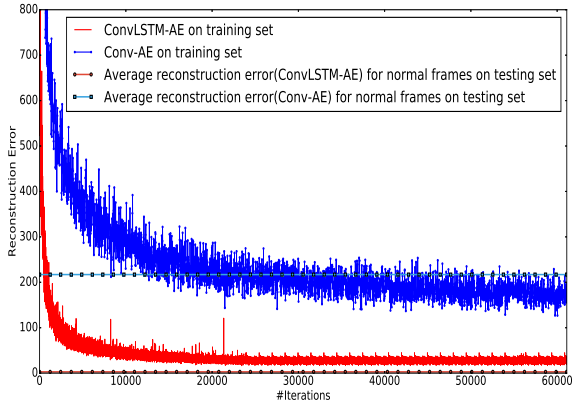


Fig. 3: The reconstruction error of ConvLSTM-AE and Conv-AE on the Moving-MNIST training and testing set.

4.3. Evaluation with the Moving-MNIST Dataset

To evaluate the performance of our method for the anomaly caused by appearance or motion separately, we deploy experiments on a synthesized Moving-MNIST dataset. Specifically, we randomly choose two digits from the MNIST dataset, and put them in the center of a black image whose size is 225×225 pixels. Then in the next 19 frames, the digits move horizontally. In this way, we can get a sequence with 20 frames. In our experiments, we synthesize 10K sequences for training data and train the network. We also synthesize 3K testing sequences for each of the following scenarios: **The change in Appearance.** To mimic the anomaly caused by the appearance, in each testing sequence, 5 consecutive frames are randomly occluded by randomly inserting a 3×3 white box. **The Change in Motion.** To mimic the anomaly caused by the change of motion, in each testing sequence, all digits keep moving straightly in the first 10 frames and moving randomly in the last 10 frames.

We show the training loss of our method and Conv-AE [9] in Figure 3. We also report the average reconstruction error of different methods for the **normal** frames for the sequences caused by the change of appearance. We can see that the reconstruction error of our method is usually smaller than that of training data, however, the reconstruction error for the Conv-AE is higher than that of training data. The possible reason is that ConvAE reconstructs the whole video clip where the abnormal frames propagates the error to the normal ones. Therefore, Conv-AE is more prone to cause false alarm, as shown in Figure 2. In contrast, ConvLSTM-AE which reconstructs current and previous frames, and the abnormal frames has no effect on normal ones, thus it is unlikely to happen the false alarm case. The comparisons between our method and that of Conv-AE are shown in Table 1. As for the anomaly caused by the change of motion, as the accuracy

Table 2: AUC of different methods on all real dataset.

	Giorno et al [19] (ECCV2016)	Conv-AE[9] (CVPR2016)	ConvLSTM-AE
Avenue	0.783	0.745	**0.770**
Ped1	-	0.681	**0.755**
Ped2	-	0.811	**0.881**
Entrance	0.691	0.910	**0.933**
Exit	0.824	0.802	**0.877**

of Conv-AE is already high, the improvement of our method over Conv-AE is not as significant as that for the anomaly caused by the change of appearance.

4.4. Evaluation with Real Anomaly Detection Datasets

Different from [9], we train different models for separate datasets because the definitions of anomaly are different, even contradicts for different datasets. Actually, training different models for different datasets are also commonly used in many existing works [16][5]. Figure 4 shows that ConvLSTM-AE quickly converges to local minimums on all datasets. The performance of different methods are shown in Table 2. We can see that our ConvLSTM-AE achieves much higher AUC than Conv-AE almost on all real datasets, which proves capacity of our model for handling the appearance and motion change. In addition to that, our model also achieves better or comparable performance as that in [19]. Furthermore, we also show the score as well as the ground-truth for a video in Subway-Entrance in Figure 5. We can see that the score we achieve is usually high for normal events, and low for abnormal events, which further verifies the effectiveness of our method.

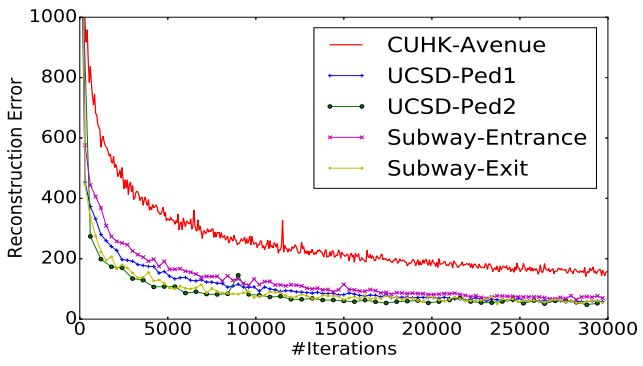


Fig. 4: The change of training reconstruction error of ConvLSTM-AE on different datasets.

5. CONCLUSION

In our paper, we propose a convolutional LSTM based Auto-Encoder framework for anomaly detection. By using CNN to encode each frame, the content of each frame can be well charaterized, and by using ConvLSTM, the motion information can be characterized. Meanwhile, ConvLSTM preserves the spatial information which helps the reconstruction of current and previous frames. Experiments on a synthesized Moving-MNIST dataset shows the robustness of ConvLSTM-AE to the change of appearance and motion. Experiments on all real datasets further show that our model greatly outperforms Convolution Auto-encoder, which demonstrates the effectiveness of our method.

Acknowledgements. This work was supported by the Shanghai Pujiang Talent Program (No.15PJ1405700), and NSFC (No. 61502304).

6. REFERENCES

[1] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *BMVC*. British Machine Vision Association, 2008, pp. 275–1.

[2] G. Willems, T. Tuytelaars, and L. V. Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *ECCV*. Springer, 2008, pp. 650–663.

[3] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IWVSPETS*. IEEE, 2005, pp. 65–72.

[4] J. Shi C. Lu and J. Jia, "Abnormal event detection at 150 fps in matlab," in *ICCV*, 2013, pp. 2720–2727.

[5] B. Zhao, F. Li, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *CVPR*. IEEE, 2011, pp. 3313–3320.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *arXiv preprint arXiv:1604.06573*, 2016.

[9] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *CVPR*, 2016.

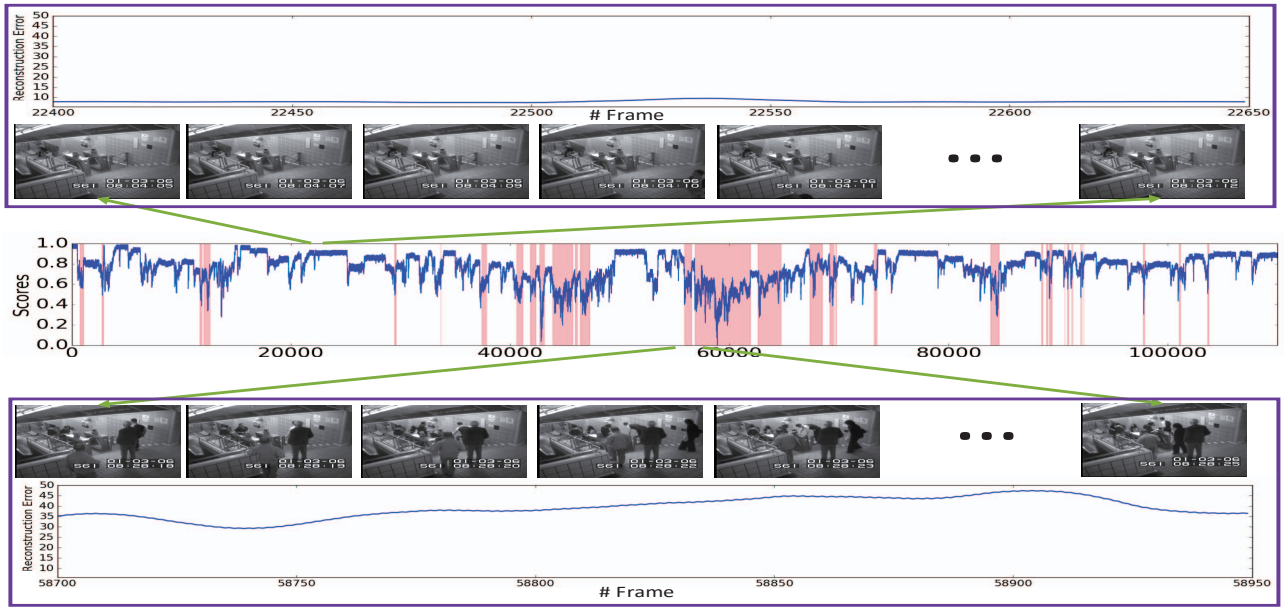


Fig. 5: The score of the whole testing video for Subway-Entrance dataset. The frames and reconstruction error in the upper purple box correspond to the ordinary moments, and frames and reconstruction error in the bottom purple box correspond to abnormal events. For an ordinary moment, the reconstruction error is lower than 10, while for an abnormal event, the reconstruction error is higher than 25.

[10] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *TPAMI*.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*. IEEE, 2015.

[12] X. Shi, Z. Chen, Hao. W, D. Yeung, D. Wong, and W. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *NIPS*, 2015, pp. 802–810.

[13] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-supervised adapted hmms for unusual event detection," in *CVPR*. IEEE, 2005, vol. 1, pp. 611–618.

[14] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognitiona review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 865–878, 2012.

[15] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[16] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.

[17] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, 2016.

[18] M. Ranzato, A. Szlam, J. Bruna, M. Mathieu, R. Collobert, and S. Chopra, "Video (language) modeling: a baseline for generative models of natural videos," *arXiv preprint arXiv:1412.6604*, 2014.

[19] A. D. Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *ECCV*. Springer, 2016, pp. 334–349.

[20] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *CVPR*, 2010, vol. 249, p. 250.

[21] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *TPAMI*, 2008.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, G. Jonathan, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM MM*. ACM, 2014, pp. 675–678.

[23] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Aistats*, 2010, vol. 9, pp. 249–256.