LEARNING APPEARANCE-MOTION NORMALITY FOR VIDEO ANOMALY DETECTION

Yang Liu¹, Jing Liu¹, Mengyang Zhao¹, Dingkang Yang¹, Xiaoguang Zhu², Liang Song^{1*}

¹Academy for Engineering & Technology, Fudan University, Shanghai, China ²SEIEE, Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Video anomaly detection is a challenging task in the computer vision community. Most single task-based methods do not consider the independence of unique spatial and temporal patterns, while two-stream structures lack the exploration of the correlations. In this paper, we propose spatial-temporal memories augmented two-stream auto-encoder framework, which learns the appearance normality and motion normality independently and explores the correlations via adversarial learning. Specifically, we first design two proxy tasks to train the two-stream structure to extract appearance and motion features in isolation. Then, the prototypical features are recorded in the corresponding spatial and temporal memory pools. Finally, the encoding-decoding network performs adversarial learning with the discriminator to explore the correlations between spatial and temporal patterns. Experimental results show that our framework outperforms the state-of-theart methods, achieving AUCs of 98.1% and 89.8% on UCSD Ped2 and CUHK Avenue datasets.

Index Terms— Video anomaly detection, unsupervised learning, memory network, generative adversarial network

1. INTRODUCTION

Video anomaly detection (VAD) is the key technology of intelligent surveillance systems, which aims to automatically detect and locate abnormal events from videos [1]. Due to the ambiguity of the definition of anomaly, the label of an event depends on the scenario. For example, fast running indoor is considered abnormal, while outdoor one is usually defined as normal. Therefore, VAD is always formulated as an out-ofdistribution detection task, which learns a model of normality that can well describe normal events. Events that cannot be described are discriminated against as abnormal. Additionally, most existing VAD methods [2–5] are unsupervised due to the difficulty of collecting abnormal events and obtaining precise frame-level labels. Earlier methods, such as one-class support vector machine, use normal samples to train a oneclass classifier that attempts to find a hyperplane as the boundary of normal events, while events whose features fall outside the boundaries are identified as anomalies. Such methods will suffer from the curse of dimensionality.

Recently, motivated by the success of deep learning in video understanding, various unsupervised deep VAD methods [6–8] have been proposed. They use only normal videos to train deep convolutional neural networks (CNNs) to solve a proxy task, such as reconstructing input video clips [2,4] or predicting future frames [3,9]. The assumption is that models learned on normal events cannot compress abnormal events that have not been seen before, leading to large reconstruction or prediction errors. Therefore, anomalies can be discriminated by measuring the deviation between the testing samples and the learned model.

Video anomalies typically include appearance and motion anomalies, corresponding to semantic shifts of information in the spatial and temporal dimensions, respectively. Thus, the key to unsupervised VAD is to learn the prototypical appearance and motion patterns of normal events. However, most existing methods [3, 4, 10] do not consider the difference of spatial and temporal features in isolation. In contrast, they train deep CNNs to learn spatial-temporal features by performing a single proxy task, lacking the exploration of unique appearance and motion patterns. Although some methods adopt reconstruction and prediction tasks simultaneously [11], they still do not treat appearance and motion anomalies differently. In recent years, two-stream structures [7, 8, 12, 13] have been proposed to separately learn appearance and motion patterns. For example, Chang et al. [8] trained two auto-encoders to obtain appearance and motion codings and used clustering algorithms to determine the boundaries of normal events. Li et al. [14] proposed cascaded auto-encoders to perform frames reconstruction and optical flow reconstruction tasks separately. Then, the appearance anomaly score and motion anomaly score are calculated independently. Most two-stream structures lack the exploration of the correlation between appearance and motion features.

In this paper, we propose spatial-temporal memories augmented two-stream auto-encoder (STM-AE) framework, which learns appearance and motion normality in isolation and explores the correlations via a prediction task. In the training phase, the appearance and motion features are extracted by two parallel auto-encoders, which are pre-trained by the denoising and optical flow generation proxy tasks.

^{*}Corresponding author. This work is supported by the Shanghai Key Research Laboratory of NSAI.

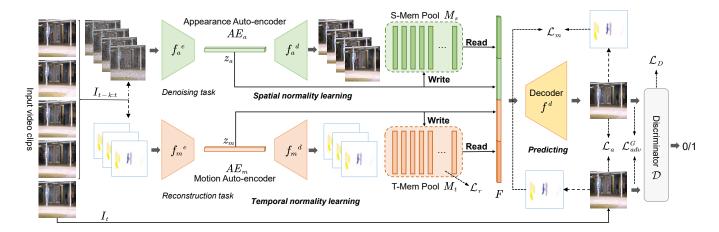


Fig. 1. Overview of the proposed saptial-temporal memories augmented two-stream auto-encoder framework.

Then, the prototypical patterns are written into the corresponding memory pools through a top-k attention-based updating mechanism. Finally, we retrieve memory items and feed the aggregated features to the decoder to predict future frames. To improve the predicting ability of the decoder, we introduce a patch discriminator [15] for adversarial learning. In the testing phase, the spatial and temporal features of testing videos are rewritten by the learned memory pools so that the prediction results are close to normal videos. Therefore, abnormal events will lead to large prediction errors. The main contributions are summarised as follows:

- We propose spatial-temporal memories augmented auto-encoder framework to learn and record prototypical appearance and motion patterns independently. Besides, we introduce adversarial learning to explore the correlations between appearance and motion normality.
- To record the prototypical patterns, we propose top-k
 attention to update the memory pools. Only part of the
 items is rewritten in each update step to ignore the personalized patterns of normal events.
- Experimental results demonstrate that the proposed framework outperforms the state-of-the-art method with AUCs of 98.1%, 89.8%, and 73.8% on UCSD Ped2, CUHK Avenue, and ShanghaiTech, respectively.

2. METHOD

2.1. Architecture

As shown in Figure 1, the proposed STM-AE framework consists of a two-stream structure, where the appearance autoencoder AE_a is used for learning spatial features and motion auto-encoder AE_m for temporal features. In the training phase, we first use normal vides to pre-train AE_a and AE_m . At time t, the AE_a compresses the input RGB images $I_{t-k:t}$

into low-dimensional appearance features z_a by performing a denoising task, where k denote the temporal length of input clips. In contrast, the AE_m obtain the motion features z_m by performing an optical flow reconstruction task. Unlike other two-stream structures, we replace the images reconstruction with the denoising task for z_a . In contrast, the denoising task requires the encoder to fully understand the spatial information since AE_m needs to reason out the missing parts (noise-masked regions) from the given parts (noise-free regions). The optical flow frames can reflect the motion information while ignoring background, so that we use the optical flow frames reconstruction as the proxy task of AE_m . The input optical flow frames are calculated by the Flownet [16].

The two auto-encoders share the same structure, consisting of a 5-layer encoder and a 6-layer decoder. Each layer is a 3×3 convolutional layer followed by batch normalization and non-linear activation. To ensure the diversity of z_a and z_m , we use the LeakyReLU as a non-linear activation layer of the encoder to retain negative values. In the pre-training phase, the goal of AE_a and AE_m is to learn how to compress appearance and motion patterns. We set the objective function for the denoising task to be minimizing the L_2 distance between the generated frames and ground truth, as follows:

$$\min_{\theta_a^e, \theta_a^d} \| f_a^d \left(f_a^e \left(\tilde{\boldsymbol{I}}_{t-k:t}; \theta_a^e \right); \theta_a^d \right) - \boldsymbol{I}_{t-k:t} \|_2^2$$
 (1)

where $I_{t-k:t}$ donetes the video clips with 20% salt and pepper noise. θ_a^e and θ_a^d denote the parameter of the appearance encoder f_a^e and decoder f_a^d . Similarly, the objective function for the reconstruction task is to optimise motion encoder f_m^e and decoder f_m^d by minimising the L_2 distance between the reconstructed optical flow and input ones, as follows:

$$\min_{\theta_{m}^{e}, \theta_{m}^{d}} \| f_{m}^{d} \left(f_{m}^{e} \left(f \left(\boldsymbol{I}_{t-k:t} \right) ; \theta_{m}^{e} \right) ; \theta_{m}^{d} \right) - f \left(\boldsymbol{I}_{t-k:t} \right) \|_{2}^{2}$$
 (2)

where $f(\cdot)$ denote the Flownet. After pre-training, θ_a^d and θ_m^d are frozen. z_a and z_m are written to the corresponding mem-

ory pools as prototype patterns of normal events. In addition, z_a and z_m are used as queries to retrieve the relevant memory items. We aggregate the raw features and retrieved items and feed them to the decoder to predict future frames. Since the memory pool is updated via normal events only during the training phase, the aggregated features F will be close to the representation of normal events in the feature space. In the testing phase, the F of the abnormal event will deviate from the direct concatenation of its z_m and z_a , which in turn leads to large prediction errors. To improve the quality of the predicted frames, we measure the difference of the generated frames in pixels and optical flow to ensure the completeness of the appearance and motion information.

2.2. Appearance and Motion Normality Recording

The spatial and temporal memory pools are used to record the prototypical appearance and motion pattern of normal events, respectively. Both the two pools contains N memory items of dimension C, denoted by the two-dimensional matrix M_s and M_t . Since the two pools share the same updating mechanism, for simplify, only the learning process of spatial normality is presented here. The write operation aims to writing z_a to M_s . The prototypical spatial pattern of normal events is recorded in M_s by updating the items in the training phase. Firstly, we flatten the z_a along the spatial dimension and obtain N query vectors of dimension C, where C equals the number of channels of z_a . The size of appearance features are $H \times$ $W \times C$, so $\hat{N} = H \times W$. We treat the memory items as linear combinations of the queries. In each update step, the weighted queries are written to the history memory items as follows:

$$\hat{\boldsymbol{m}}_{c}^{i} = q \left(\boldsymbol{m}_{c}^{i} + \boldsymbol{w}_{c}^{i} \boldsymbol{Q} \right) \tag{3}$$

where $g(\cdot)$ denotes the L_2 normalization, which are used to keep the data scale of the updated memory items \hat{m}_s^i same as history memory items m_s^i . w_s^i is a weight vector of size $1 \times \hat{N}$, used to weight queries Q. Different from previous work [10], considering the diversity of normal events, only part of the query vectors are written into the memory pool instead of all, which helps the pool to record general patterns of normal events. Therefore, we only retain the weight of the top-k relevant query vectors when computing w_s^i . Specifically, We first compute the cosine similarity of m_s^i to all queries, denoted by \tilde{w}^i . Then, the top-k values of \tilde{w}^i are retained, and the others are set to 0. We perform softmax normalization on \tilde{w}^i to calculate the weights w_s^i , as follows:

$$\boldsymbol{w}_{s}^{i} = \frac{\exp\left(\tilde{\boldsymbol{w}}^{i}\right) - 1}{\sum_{j=1}^{\hat{N}} \exp\left(\tilde{\boldsymbol{w}}_{j}^{i}\right) - \hat{N}}$$
(4)

The read operation is to reconstruct the queries Q using the memory items. We treat the reconstructed query \hat{z}_a as the linear combination of the memory items. The weights are calculated by performing softmax normalization on the cosine similarity of the query to all memory items.

2.3. Adversarial Learning and Frame Prediction

To explore the correlation between appearance and motion features, we concatenate the reconstructed and raw features and feed them to the 5-layer decoder f^d for predicting future frames. We measure the quality of the predicted images at both the pixel domain and optical flow domain to ensure spatial and temporal information integrity. The appearance loss \mathcal{L}_a is defined as the L_2 distance between the predicted frame \hat{I}_t and ground truth I_t , as follows:

$$\mathcal{L}_a = \parallel \hat{\boldsymbol{I}}_t - \boldsymbol{I}_t \parallel_2^2 \tag{5}$$

Similarity, we define motion loss \mathcal{L}_m as follows:

$$\mathcal{L}_{m} = \| f(\hat{\mathbf{I}}_{t}, \mathbf{I}_{t-1}) - f(\mathbf{I}_{t}, \mathbf{I}_{t-1}) \|_{2}^{2}$$
 (6)

In addition, to further improve the ability to fuse and decode appearance and motion features, We treat the two encoders f_a^e and f_m^e and the decoder f^d as the generator \mathcal{G} . A patch discriminator \mathcal{D} [15] is introduced to perform adversarial learning with \mathcal{G} . The discriminator accepts image patches as input and discriminates whether the input is from real future frames (output 0) or predicted ones (output 1). In contrast, the objective of \mathcal{G} is to generate frames that can be discriminated against as real ones by the discriminator. The adversarial loss for training the generator, denoted by \mathcal{L}_{adv}^G , is as follows:

$$\mathcal{L}_{adv}^{G} = \sum_{i,j} \frac{1}{2} \left(\mathcal{D}(\hat{\boldsymbol{I}}_{t}^{i,j}) - 1 \right)^{2} \tag{7}$$

where i, j are the spatial index of patches. The loss for training the discriminator, denoted by \mathcal{L}_D , is defined as follows:

$$\mathcal{L}_D = \sum_{i,j} \frac{1}{2} \left(\mathcal{D}(\boldsymbol{I}_t^{i,j}) - 1 \right)^2 + \sum_{i,j} \frac{1}{2} \mathcal{D}(\hat{\boldsymbol{I}}_t^{i,j})^2 \qquad (8)$$

Additionally, we introduce \mathcal{L}_r to ensure the representativeness of the memory items, as follows:

$$\mathcal{L}_r = \sum_{i=1}^{N} \| \mathbf{q}^i - \mathbf{m}_1^i \|_2^2 - \| \mathbf{q}^i - \mathbf{m}_2^i \|_2^2$$
 (9)

where m_2^i and m_2^i denote the first and second nearest memory items to query q^i . Balanced by λ_1 , λ_2 and λ_3 , the total loss for training the generator, denoted by \mathcal{L}_G , is as follows:

$$\mathcal{L}_G = \mathcal{L}_a + \lambda_1 \mathcal{L}_m + \lambda_2 \mathcal{L}_r + \lambda_3 \mathcal{L}_{adv}^G$$
 (10)

2.4. Anomaly Score

After training, we assume that the prototypical appearance and motion features of normal events have been recorded in two memory pools. The generator can efficiently compress and predict normal events so that the prediction error is relatively low. In contrast, the aggregated features of abnormal events will be close to normal events, which leads to

poor prediction results. Therefore, we obtain the anomaly score by calculating the prediction error e. Follow previous works [3, 10], e is defined as the peak signal-to-noise ratio between the predicted frames \hat{I}_t and the ground truth I_t , as follows:

$$e_t = 10 \log_{10} \frac{255^2}{\|\hat{I}_t - I_t\|_2^2}$$
 (11)

Finally, we use maximum-minimum normalization to map e_t to anomaly score s_t in the range of [0, 1], as follows:

$$s_t = \frac{e_t - \min_t e_t}{\max_t e_t - \min_t e_t} \tag{12}$$

3. EXPERIMENTS

3.1. Implementation Details

Datasets. We conduct comparative experiments and ablation studies on three standard benchmarks to validate the proposed framework, which are introduced below:

- UCSD Ped2 [17] is a medium-scale VAD dataset containing 16 training videos and 12 testing videos. All videos were captured from outdoor scenes with a camera view parallel to the street. The anomalies include biking, skateboarding, and driving on the sidewalk.
- CUHK Avenue [18] is a large-scale dataset with 21 training videos and 16 testing videos. There are 47 abnormal events, such as unusual running and loitering.
- ShanghaiTech [2] is a large-scale and challenging dataset for unsupervised VAD, including 330 training videos and 107 testing videos. It contains 130 abnormal events from 13 different scenes.

Evaluation metric. VAD is a regression task aiming to calculate an anomaly score in the range of [0,1]. The score for a normal frame should be close to 0, while that of an abnormal one should be 1. We calculate true positive rates and false-positive rates under numerous thresholds. The frame-level area under the curve (AUC) of the receiver operation characteristic is used as an evaluation metric.

Training details. The STM-AE framework is trained using the Pytorch framework with an Nvidia Geforce RTX 2080Ti GPU. Adam is used as the optimizer with a batch size of 8. The initial learning rate is set to 4×10^{-4} and is decayed by the cosine annealing methods. All video frames are firstly resized to 256×256 pixels. The framework takes four successive frames as input and predicts the next frame. We set N and k to 32 and 8. The trade-off parameters λ_1 , λ_2 and λ_3 are set to 0.6, 0.01 and 0.05, respectively.

3.2. Quantitative Comparison

We quantitatively compared the frame-level AUC of the proposed STM-AE framework with existing unsupervised

Table 1. Results of the frame-level AUC comparison. Bolded numbers indicate the best performance, and underlined ones indicate the second-best performance.

Туре		Method	Frame-level AUC (%)					
		1,10,110	UCSD Ped2	CUHK Avenue	ShanghaiTech			
>		Kim et al. [19]	69.3	-	-			
Shadow		Lu et al. [18]	-	80.9	-			
		Xu et al. [20]	90.8	-	-			
		Tudor et al. [21]	82.2	80.6	-			
Deep learning-based	Single-task	Luo et al. [2]	92.2	81.7	68.0			
		Gong et al. [4]	94.1	83.3	71.2			
		Zhang et al. [6]	95.4	86.8	73.6			
		Park et al. [10]	97.0	88.5	72.8			
		Ye et al. [9]	96.8	86.2	73.6			
		Liu et al. [3]	95.4	85.1	72.8			
	Two-stream	Cai <i>et al</i> . [7]	96.6	86.6	73.7			
		Chang et al. [8]	96.5	86.0	73.3			
		Doshi et al. [12]	97.8	86.4	71.6			
		Yu et al. [13]	97.3	<u>89.6</u>	74.6			
		STM-AE (Ours)	98.1	89.8	73.8			

VAD methods on the UCSD Ped2 [17], CUHK Avenue [18] and ShanghaiTech [2] datasets. The results are reported in Table 1. The methods involved in the comparison include traditional shallow methods [18–21], methods with single proxy task (single-task) [2–4, 6, 9, 10] and two-stream structure-based (two-stream) methods [7,8,12,13]. Our STM-AE framework outperforms the state-of-the-art methods on UCSD Ped2 and CUHK Avenue datasets, achieving AUCs of 98.1% and 89.8%, respectively, which are significantly higher than shallow and single-task methods. The two-stream methods generally outperform single-based methods, indicating that separate consideration of spatial normality and temporal normality is effective to address unsupervised VAD. Further, our STM-AE framework has explored the correlations between spatial and temporal features.

On the ShanghaiTech dataset, the STM-AE framework achieves the second-best result, slightly lower than [13] by 0.8%. A possible reason is that our spatial-temporal memories cannot adapt to data from different scenes. Compared to the single-scene of UCSD Ped2 and CUHK Avenue, ShanghaiTech includes videos from 13 different scenes. Additionally, the STM-AE framework takes 0.28 seconds on average to calculate the anomaly scores of a given frame, i.e., the inference speed is around 40 fps, satisfying the needs of real-time detection in real-world applications.

3.3. Ablation Studies

To verify the effectiveness of joint normality learning, spatial-temporal memories, and training loss, we conducted ablation studies on the UCSD Ped2 dataset. The results are presented in Table 2, and the discussion is below.

Effectiveness of appearance and motion normality learning. To demonstrate the advantages of joint learning

Table 2. Results of ablation studies. We report frame-AUC on the UCSD Ped2 dataset. Bolded number indicates the best performance, and underlined one indicates the second-best.

Model	AE_a	M_s	AE_m	M_t	\mathcal{L}_a	\mathcal{L}_m	\mathcal{L}_r	\mathcal{L}_{adv}	AUC (%)
1	✓	✓			√		√	✓	96.8
2			\checkmark	\checkmark		\checkmark	\checkmark	\checkmark	93.6
3	✓		\checkmark		✓	\checkmark		\checkmark	95.5
4	✓	√	✓	√	√	√	√	✓	98.1
5	✓	✓	✓	✓	√		√		96.9
6	✓	\checkmark	\checkmark	\checkmark	✓	\checkmark	\checkmark		97.4
7	✓	\checkmark	\checkmark	\checkmark	✓		\checkmark	\checkmark	<u>97.5</u>
8	✓	\checkmark	\checkmark	✓	✓	\checkmark		\checkmark	96.3

of appearance and motion normality over single normality learning, we compared the performance of models with different component combinations. The joint normality (model 4) achieves a 1.3% and 4.5% AUC improvement compared to single appearance normality (model 1) and single motion normality (model 2), respectively, demonstrating the effectiveness of the two-stream structure. For unsupervised VAD, the appearance normality seems more critical than the motion normality. Additionally, Compared to model 4, model 3 remove the spatial-temporal memories and directly concatenate the features from f_a^e and f_m^e during the decoding process. The 2.6% AUC gap demonstrates that the spatial-temporal memories can store prototypical features of normal events and effectively improve the performance of unsupervised VAD tasks.

Effectiveness of training loss. Models 5-7 explore the effectiveness of training loss, and the results showed that simultaneous constraints on the completeness of appearance and motion information are meaningful. Comparing the performance of model 7 with that of model 4, we find that the patch discriminator brings a 0.6% AUC gain, indicating that adversarial learning can improve the ability to fuse and decode appearance and motion features.

Sensitivity to N and k. In addition, we explore the effect of the number of memory items in the memory pool and the number of selected memory items in the write operations, i.e., the sensitivity of the STM-AE framework to the hyperparameters N and k. The results are shown in Figure 2. As the value of N increases, the AUC with different k rises first and then falls. A small memory pool will lose information due to insufficient capacity. In contrast, a large memory pool will record additional worthless features that cannot represent the prototypical pattern of normal events. The curves with fixed k (2, 4, 8) are generally above the curve with no constraint on k values (marked by triangles), indicating that updating only top-k relevant memory items instead of all help the memories to learn prototypical spatial and temporal patterns of normal events. Qualitatively, our updating mechanism brings a 0.9% AUC gain on the UCSD Ped2 dataset.

3.4. Visual Results

In Figure 3, we show the spatial localization results of the STM-AE framework for abnormal events. The experiments

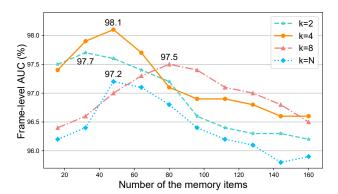


Fig. 2. Results of ablation studies on the sensitivity.

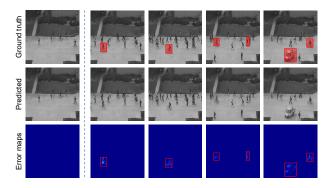


Fig. 3. Results of spatial localization of abnormal events.

are conducted on the UCSD Ped2 [17] dataset. Rows 1-3 are the ground-truth future frames, predicted frames, and prediction errors. The first column is for normal walking, while columns 2-4 are for abnormal events, including riding, skateboarding, and driving on the sidewalk. The regions marked by red boxes are the locations where abnormal events occur. Comparing columns 1 and 2-4, we find that the STM-AE framework can predict normal events effectively with minor errors, especially for the background parts. For abnormal frames, the prediction results for regions containing abnormal events are significantly worse than for normal motion and background areas, indicating that the STM-AE framework can localize unusual appearance and motion.

4. CONCLUSION

In this paper, we address unsupervised VAD by considering appearance anomalies and motion anomalies separately. The proposed STM-AE framework learns prototypical appearance and spatial patterns separately and records them in spatial-temporal memories. And the intrinsic correlation between appearance and motion normality is investigated via adversarial learning. Experimental results demonstrate the effectiveness of joint normality learning, and the STM-AE framework outperforms the state-of-the-art methods. For the problem of per-

formance degradation in processing multi-scene videos, we will explore hierarchical memories to improve the adaptability to multi-scene data in future work.

5. REFERENCES

- [1] Yang Liu, Jing Liu, Mengyang Zhao, Shuang Li, and Liang Song, "Collaborative normality learning framework for weakly supervised video anomaly detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2022.
- [2] Weixin Luo, Wen Liu, and Shenghua Gao, "A revisit of sparse coding based anomaly detection in stacked rnn framework," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 341–349.
- [3] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao, "Future frame prediction for anomaly detection—a new baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6536–6545.
- [4] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vi*sion, 2019, pp. 1705–1714.
- [5] Yang Liu, Jing Liu, Jieyu Lin, Mengyang Zhao, and Liang Song, "Appearance-motion united auto-encoder framework for video anomaly detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1–1, 2022.
- [6] Yu Zhang, Xiushan Nie, Rundong He, Meng Chen, and Yilong Yin, "Normality learning in multispace for video anomaly detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.
- [7] Ruichu Cai, Hao Zhang, Wen Liu, Shenghua Gao, and Zhifeng Hao, "Appearance-motion memory consistency network for video anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, pp. 938–946.
- [8] Yunpeng Chang, Zhigang Tu, Wei Xie, and Junsong Yuan, "Clustering driven deep autoencoder for video anomaly detection," in *European Conference on Computer Vision*. Springer, 2020, pp. 329–345.
- [9] Muchao Ye, Xiaojiang Peng, Weihao Gan, Wei Wu, and Yu Qiao, "Anopen: Video anomaly detection via deep predictive coding network," in *Proceedings of the 27th* ACM International Conference on Multimedia, 2019, pp. 1805–1813.
- [10] Hyunjong Park, Jongyoun Noh, and Bumsub Ham, "Learning memory-guided normality for anomaly detection," in *Proceedings of the IEEE/CVF Conference*

- on Computer Vision and Pattern Recognition, 2020, pp. 14372–14381.
- [11] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua, "Spatio-temporal autoencoder for video anomaly detection," in *Proceedings of the 25th ACM International Conference on Multimedia*, 2017, pp. 1933–1941.
- [12] Keval Doshi and Yasin Yilmaz, "Continual learning for anomaly detection in surveillance videos," in *Proceed*ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 254–255.
- [13] Guang Yu, Siqi Wang, Zhiping Cai, En Zhu, Chuanfu Xu, Jianping Yin, and Marius Kloft, "Cloze test helps: Effective video anomaly detection via learning to complete video events," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 583–591.
- [14] Nanjun Li, Faliang Chang, and Chunsheng Liu, "Spatial-temporal cascade autoencoder for video anomaly detection in crowded scenes," *IEEE Transactions on Multimedia*, vol. 23, pp. 203–215, 2020.
- [15] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2794–2802.
- [16] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.
- [17] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 1, pp. 18–32, 2013.
- [18] Cewu Lu, Jianping Shi, and Jiaya Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2720–2727.
- [19] Jaechul Kim and Kristen Grauman, "Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates," in 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009, pp. 2921–2928.
- [20] Dan Xu, Yan Yan, Elisa Ricci, and Nicu 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.
- [21] Radu Tudor Ionescu, Sorina Smeureanu, Bogdan Alexe, and Marius Popescu, "Unmasking the abnormal events in video," in *Proceedings of the IEEE International* Conference on Computer Vision, 2017, pp. 2895–2903.