Pattern Recognition 138 (2023) 109335

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Memory-augmented appearance-motion network for video anomaly detection

Le Wang[a], Junwen Tian[a], Sanping Zhou[a,*], Haoyue Shi[a], Gang Hua[b]

[a] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China
[b] Wormpex AI Research, Bellevue, WA 98004, USA

ARTICLE INFO

Article history:
Received 2 December 2021
Revised 21 December 2022
Accepted 12 January 2023
Available online 15 January 2023

Keywords:
Anomaly detection
Memory network
Autoencoder
Abnormal events

ABSTRACT

Video anomaly detection is a promising yet challenging task, where only normal events are observed in the training phase. Without any explicit classification boundary between normal and abnormal events, anomaly detection can be turned into an outlier detection problem by regarding any event that does not conform to the normal patterns as an anomaly. Most of the existing works mainly focus on improving the representation of normal events, while ignore the relationship between normal and abnormal events. Besides, the lack of restrictions on classification boundaries also leads to performance degradation. To address the above problems, we design a novel autoencoder-based Memory-Augmented Appearance-Motion Network (MAAM-Net), which consists of a novel end-to-end network to learn appearance and motion feature of a given input frame, a fused memory module to build a bridge for normal and abnormal events, a well-designed margin-based latent loss to relieve the computation costs, and a pointed Patch-based Stride Convolutional Detection (PSCD) algorithm to eliminate the degradation phenomenon. Specifically, the memory module is embedded between the encoder and decoder, which serves as a sparse dictionary of normal patterns, therefore it can be further employed to reintegrate abnormal events during inference. To further distort the reintegration quality of abnormal events, the margin-based latent loss is leveraged to enforce the memory module to select a sparse set of critical memory items. Last but not least, the simple yet effective detection method focuses on patches rather than the overall frame responses, which can benefit from the distortion of abnormal events. Extensive experiments and ablation studies on three anomaly detection benchmarks, i.e., UCSD Ped2, CUHK Avenue, and ShanghaiTech, demonstrate the effectiveness and efficiency of our proposed MAAM-Net. Notably, we achieve superior AUC performances on UCSD Ped2 (0.977), CHUK Avenue (0.909), and ShanghaiTech (0.713). The code is publicly available at https://github.com/Owen-Tian/MAAM-Net.

1. Introduction

Video anomaly detection aims to identify abnormal events that do not conform to normal event patterns [1–3]. It exhibits significant importance and necessity when applied in video surveillance [4,5], and thus has received widespread attention from the community in recent years.

Since abnormal events rarely occur in real-life scenarios and also manual labeling is labor-intensive and time-consuming, it is extremely difficult to collect enough training samples of abnormal events. Therefore, it is intractable for conventional classification methods to handle the anomaly detection problem due to severely imbalanced samples. Naturally, it necessitates the model to be able to identify anomalies solely based on normal events as supervision. Existing works [6,7] address this problem mainly by training a network that can leverage normal events to portrait the feature distribution of normal events. During the inference phase, the distance between the unknown inputs and the learned distribution becomes the key criterion to identify anomalies. According to the number of input frames, these works can be divided into two categories: 1) reconstruction-based methods [8,9] take a single frame as input and output the corresponding reconstruction result, and the reconstruction error is used to identify anomalies; 2) prediction-based methods [10,11] take multiple previous frames to predict the subsequent frame or directly predict the optical flow from a single frame, where the prediction error, i.e., the difference between the prediction and the ground-truth, is exploited to compute the anomaly score.

* Corresponding author.
E-mail addresses: lewang@xjtu.edu.cn (L. Wang), tianjunwen@stu.xjtu.edu.cn (J. Tian), spzhou@xjtu.edu.cn (S. Zhou), shyern@stu.xjtu.edu.cn (H. Shi).

https://doi.org/10.1016/j.patcog.2023.109335
0031-3203/

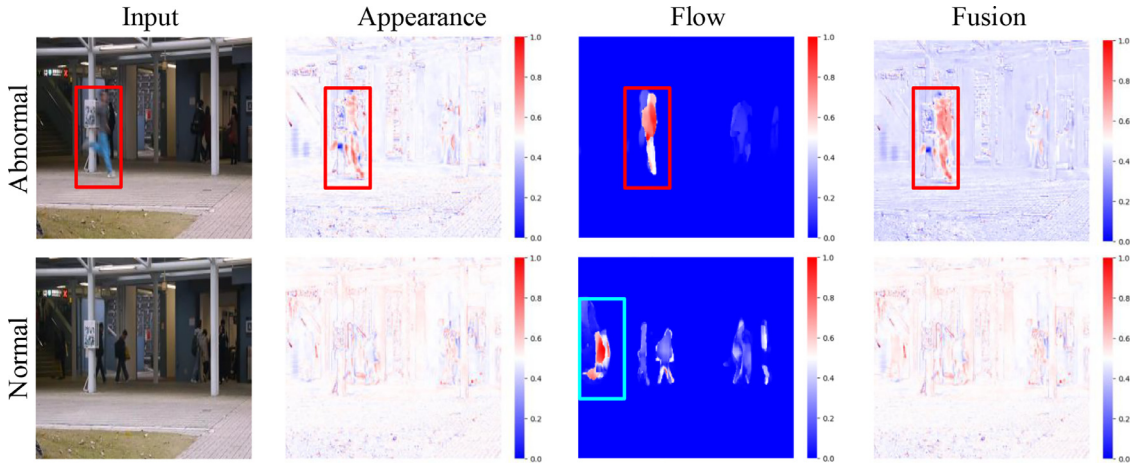


Fig. 1. Visualization of some detection results. The four columns are the original input, normalized appearance reconstruction difference, flow prediction difference and fusion difference, respectively. The difference is normalized to [0,1]. Abnormal events are enclosed by red box and false alarms are enclosed by green box. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The majority of current methods [12,13] adopt the autoencoder framework [9] to reconstruct or predict frames from the training samples. During inference, abnormal events are expected to incur a much larger error than normal events. However, this assumption does not always hold in practice. The absence of constraints on the relationship between normal and abnormal events will inevitably lead to missing detections. What's worse, the lack of boundary restriction makes the model prone to misclassify hard positive samples from the true negatives, e.g., distinguish a skateboarder from a quick walker. These ambiguities severely degrade the anomaly detection accuracy.

This paper presents a novel hybrid network, Memory-Augmented Appearance-Motion Network (MAAM-Net), which simultaneously utilizes frame reconstruction and flow prediction to tackle the aforementioned challenges. For the relationship between normal and abnormal events, the memory module in the proposed MAAM-Net uses the normal events to reintegrate unknown events. Since abnormal events usually exhibit anomalous shapes and speeds, it is difficult to be integrated with normal events. Thus the memory module can act as a destroyer for abnormal events and a re-integrator for normal events in the feature space.

For the challenge of lacking boundary restriction, we further introduce a new margin-based latent loss, where we set a margin between the differences of the encoded feature and its reintegration. The latent loss forces the memory module to choose the minimal yet crucial items for the reintegration. Thus, the computation costs brought by the dissimilar items decrease significantly. Meanwhile, the abnormal events are much more difficult to reintegrate with fewer items, incurring a much larger reconstruction error.

What's more, since the abnormal events usually appear in local regions, we argue that the previous frame-level detection methods may miss an abnormal event occurring within a small image region. As illustrated in Fig. 1, we can see that the abnormal region (enclosed by the red box) is much smaller than the entire frame. Thus, we propose a patch-based detection method that uses the maximal local generative error from the fusion of the appearance and motion branches as the anomaly score of a given frame.

Our MAAM-Net is composed of an encoder, a memory module, an appearance decoder and a motion decoder, where the memory module is embedded between the encoder and each of the decoders as a sparse dictionary to store the diverse patterns of normal events. Given a video frame, the frame features are extracted by the encoder. Then, we reintegrate the encoded features by fusing them with the similarity weighted items of the memory module. Finally, the reintegrated feature will be fed into the appearance and motion decoders to reconstruct the input frame and predict the corresponding optical flow. In the testing phase, the proposed patch-based detection method can jointly generate an appearance error map and a motion error map, which are then fused to generate the final anomaly score. As shown in Fig. 1, the fusion results of appearance and motion branches can highlight the abnormal region (in the first row) and eliminate the possible false detection of a single branch (in the second row).

We conduct extensive experiments and ablation studies on three benchmarks: i.e., UCSD Ped2, CUHK Avenue, and ShanghaiTech datasets. The state-of-the-art results demonstrate the effectiveness and efficiency of our method. To the best of our knowledge, this is the first work that combines the appearance-motion autoencoder with a memory module for video anomaly detection. In summary, the main contributions of this paper can be highlighted as follows:

- A memory-augmented appearance-motion network (MAAM-Net) is proposed for video anomaly detection.
- A margin-based latent loss is introduced to improve the sparsity and generalization of normal patterns in the memory module, ensuring the high contrast between normal and abnormal events while reducing the computational cost.
- A patch-based detection method is proposed to highlight the local response of abnormal events and suppress the response of normal events.
- Extensive experimental results compared with competing methods and ablative studies validate the efficacy of our proposed method.

The rest of this paper is organized as follows. Section 2 reviews related works in anomaly detection and memory network. In Section 3, we present the MAAM-Net in detail, including the problem formulation, the network architecture, the loss functions, and the patch-based detection method. The experimental results and discussions are presented in Section 4. Section 5 draws the conclusion. Section 6 presents the limitation of our method and future work.

2. Related work

2.1. Anomaly detection

Most previous works [12,14] define anomaly detection as an unsupervised problem since data of abnormal events are unavailable

in the training stage. They mainly adopt generative methods to detect anomalies by fully learning the regularities of normal events. Plenty of works take advantage of the generative adversarial network (GAN) [9] framework and use an auxiliary discriminator to play the role of the novelty detector [6,8,9,11,12,14–16]. Doshi et al. [15] propose an online method in surveillance videos with asymptotic bounds on the false alarm rate. Sabokrou et al. [9] take advantage of the denoising-GAN by using the noisy input frames. AbnormalGAN [12] reverses the generative target by using the motion feature to reconstruct the appearance feature and vice versa. GANomaly [14] uses an extra CNN to encode the reconstructed frame to correct the reconstruction. Furthermore, Perera et al. [8] use two extra discriminators, i.e., a visual discriminator and a latent discriminator, to supervise the reconstruction. Chen et al. [16] introduce a noise-modulated adversarial learning method. Acsintoae et al. [17] merge a self-supervised multi-task model with cycleGAN to tackle this problem. However, the auxiliary discriminator brings instability and additional computation costs to the training stage. In addition, Liu et al. [11] propose a future frame prediction framework which uses multiple previous frames to predict the subsequent single one. Nguyen et al. [6] simplify the previous work and integrate the model into a united framework. They try to stabilize the GAN training process by adding more supervision. Motivated by Liu et al. [11] and Nguyen et al. [6], we also regard the motion feature as an important component.

Moreover, as a few works [1,2,7,10,18,19] notice that a single feature distribution is insufficient to describe the patterns of various normal events, they argue that the model needs to better portrait the diversity of normal events. Abati et al. [18] use the normal events to train an autoregressive density estimation network. They need multiple frames as the input to train the network. Object-centric autoencoder [7] directly uses the one-versus-rest strategy to classify normal events in feature space, the key difference is that the input is not the whole frame, but the object extracted from the frame in advance. Luo et al. [1] introduce the sparse coding and embed it into sRNN [20] to learn a dictionary for normal events. Yu et al. [21] propose a localization based reconstruction model with a self-paced refinement scheme to detect anomalies, and Wang et al. [22] design a pretext task, i.e., solving spatio-temporal jigsaw puzzles, while they both need to extract a large number of objects of interest in advance. What's more, MemAE [2] introduces a memory module to store the sparse features of normal events and use them to reintegrate the anomalies, which needs 3D convolution layers. Park et al. [10] significantly decrease the capacity of the memory module, but they require to manually update the parameters of the memory module. Moreover, Cai et al. [19] propose an appearance-motion memory consistent network to model the consistency between the appearance and motion of regular videos, and Liu et al. [23] design the network of multi-level memory modules in an autoencoder with skip connections to memorize normal patterns for optical flow reconstruction, while they both require a sequence of frames as the input. The difference in the feature space thus becomes an additional criteria to improve the performance.

However, the main weakness of the above methods is the imbalance between inference speed and accuracy. The work most related to ours is MemAE [2], which stores the normal patterns through a memory module. It employs 3D convolutional layers to find the internal correlation of temporal sequence, while we simply use the optical flow features to describe the motion information with 2D convolutional operation. Moreover, MemAE [2] does not apply restrictions on the latent features, leading to large amount parameters and poor generalization ability, instead we introduce a margin-based latent loss to force the memory module to select minimal yet crucial items for reintegration, and thus can improve the generalization. Owing to the aforementioned two distinctions, we achieve higher performance with a faster inference speed than MemAE [2].

2.2. Memory module

Memory network is initially introduced by Weston et al. [24]. It avoids the weakness of LSTM [25] that cannot remember long-term features. However, when the number of training samples grows significantly, we need to maintain a much bigger memory which will limit the inference speed, and we argue that the memory module will become a simple storage module if the update process involves human operation. As for the anomaly detection works where the memory is introduced, Luo [13] and Park et al. [10] use a manual update for the memory module, and they use carefully selected hyper-parameters to train the network. Inspired by MemAE [2], we apply the memory module to our model and further introduce a new margin-based latent loss to restrict the feature reintegration in the feature space.

3. Proposed method

In this section, we formulate the anomaly detection task and describe the proposed MAAM-Net in detail. As presented in Fig. 2, the proposed MAAM-Net consists of four major components, i.e., an encoder, an augmented memory module, an appearance decoder, and a motion decoder. Given an original video frame $\mathbf{I}$, the encoder first extracts its feature $\mathbf{z}$. The memory module reintegrates a feature vector $\hat{\mathbf{z}}$ by retrieving the most relevant items in the memory $\mathcal{M}$. Then, the reintegrated feature vector $\hat{\mathbf{z}}$ is passed to the appearance decoder and the motion decoder for video frame reconstruction and optical flow prediction, respectively. The final outputs are the reconstructed frame $\hat{\mathbf{I}}$ and the corresponding predicted optical flow $\hat{\mathbf{F}}$. During training, the encoder and two decoders are jointly optimized to minimize the generative errors between the input and each of the corresponding outputs. During inference, the patch-level weighted summation errors of L_2 distance-based appearance similarity and L_1 distance-based motion similarity are fused to detect anomalies.

3.1. Problem formulation

Anomaly detection is generally regarded as an unsupervised learning task, as it is solely based on learning normal samples during training. In this paper, we tackle this problem in a generative way. Specifically, for each given input frame $\mathbf{I}$, our objective is to generate a reconstructed frame $\hat{\mathbf{I}}$ and predict its optical flow $\hat{\mathbf{F}}$ with the next frame. During inference, we aim to generate a frame-level anomaly prediction $y \in \{0, 1\}$ for each testing frame, where 0 indicates normal event and 1 denotes abnormal event. This is accomplished by assigning an anomaly score for each testing frame, and then classifying each of the testing samples by thresholding the anomaly score with a threshold τ. Here, we use the difference between the input frame $\mathbf{I}$ and the reconstructed frame $\hat{\mathbf{I}}$ together with the difference between the original optical flow $\mathbf{F}$ and the predicted optical flow $\hat{\mathbf{F}}$ to determine the anomaly score of a testing frame. Meanwhile, the spatial location of the abnormal events can be localized in the frame using the pixel-wise error responses.

3.2. Memory-augmented appearance-motion network

3.2.1. Encoder and decoders

The encoder contains five sub-blocks, and the detailed architecture is given in Table 1. Each sub-block consists of three layers: a convolutional layer, a batch normalization layer [26], and a ReLU activation [27]. After the encoding procedure $f_e(\cdot)$, we obtain an encoded feature vector $\mathbf{z} = f_e(\mathbf{I})$.

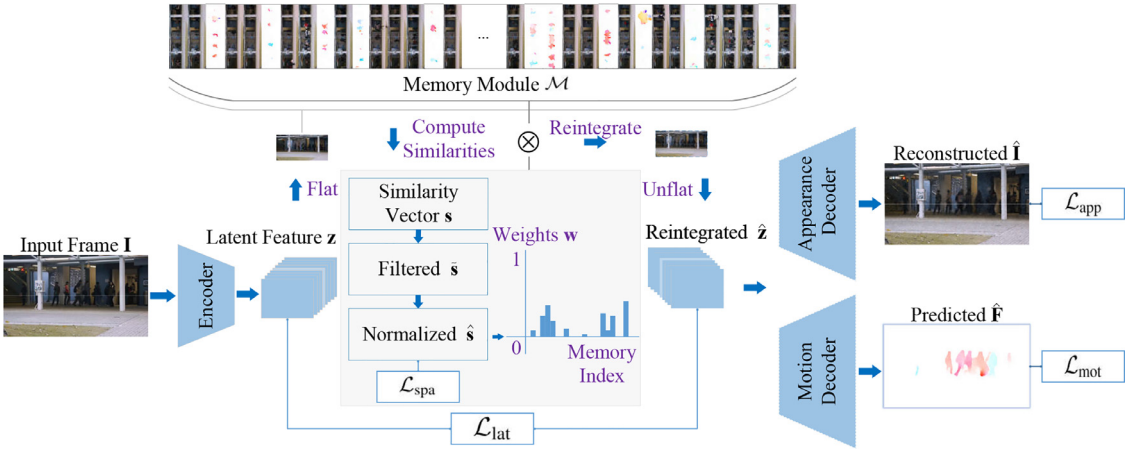


Fig. 2. Overview of our proposed MAAM-Net. It consists of fours components: an encoder, a memory module, an appearance decoder, and a motion decoder. The encoder extracts the subspace latent feature $\mathbf{z}$ from the input frame $\mathbf{I}$. The memory module $\mathcal{M}$ stores the sparse features of normal events and uses them to reintegrate $\hat{\mathbf{z}}$, according to the similarity with $\mathbf{z}$. Finally, by decoding the reintergrated $\hat{\mathbf{z}}$, the appearance decoder outputs the reconstructed frames $\hat{\mathbf{I}}$ and the motion decoder outputs an optical flow $\hat{\mathbf{F}}$ predicting the motion between I_t and I_{t+1}.

Table 1
The detailed network structure. It consists of four components, including an encoder, a memory module, an appearance decoder, and a motion decoder. N represents the memory capacity.

Stage	Type	Filters	Stride
Encoder	Conv.	$3 \times 3 \times 64$	1
		BN + ReLU	
	Conv.	$3 \times 3 \times 128$	2
		BN + ReLU	
	Conv.	$3 \times 3 \times 256$	2
		BN + ReLU	
	Conv.	$3 \times 3 \times 512$	2
		BN + ReLU	
	Conv.	$3 \times 3 \times 512$	2
		BN + ReLU	
Memory	Initialize, Train and Test	[N,512]	
App./Mot. Decoder	Deconv.	$3 \times 3 \times 256$	2
		BN + Dropout(0.3) + ReLU	
	Deconv.	$3 \times 3 \times 256$	2
		BN + Dropout(0.3) + ReLU	
	Deconv.	$3 \times 3 \times 128$	2
		BN + Dropout(0.3) + ReLU	
	Deconv.	$3 \times 3 \times 64$	2
		BN + Dropout(0.3) + ReLU	
	Conv.	$3 \times 3 \times (3/2)$	1

The appearance and motion decoders both contain a set of sub-blocks with an additional output convolutional layer, as presented in Table 1. Each sub-block contains four layers, i.e., a convolution-transpose-2d layer, a batch normalization layer [26], a dropout (dropout ratio $p_{drop} = 0.3$) layer [28], and a ReLU activation [27]. The final convolutional layer is used to reproduce the inputs by adjusting the number of output channels. Except for the difference between the number of output channels in the final convolutional layer (3 for the appearance decoder, 2 for the motion decoder), the motion decoder shares the same structure with the appearance decoder. The ground truth optical flow is extracted from adjacent video frames by FlowNet2 [29]. For each video, the last frame is excluded to ensure sample consistency. Since we use a memory module to isolate the encoding and decoding stages, the skip-connection [6] is not used, because it may impair the learning of the memory module. Taking the reintegrated $\hat{\mathbf{z}}$ as input, the appearance decoder f_{app} reconstructs the original input frame $\mathbf{I}$ as $\hat{\mathbf{I}} = f_{\text{app}}(\hat{\mathbf{z}})$, and the motion decoder f_{mot} predicts the corresponding optical flow as $\hat{\mathbf{F}} = f_{\text{mot}}(\hat{\mathbf{z}})$.

3.2.2. Memory module

The memory module is realized as a matrix $\mathcal{M} \in \mathbb{R}^{N \times C}$, where N is the memory capacity and C is the channel number of the encoded feature $\mathbf{z}$, which is used as a query to retrieve a set of similar items. The encoded feature $\mathbf{z}$ is organized as a matrix $\mathbf{z} \in \mathbb{R}^{B \times H \times W \times C}$, where B, H, and W are the batch size, height, and width of the network input after the encoder module. The memory module thus acts as a sparse dictionary to store the crucial representations of normal patterns during training. Inspired by Gong et al. [2], we use the set of similar items from the memory module to reintegrate the encoded feature through the following processes.

First, we compute the similarities (e.g., cosine similarity) between the encoded feature $\mathbf{z}$ and all memory items $\mathcal{M} = \{\mathbf{m}_i\}_{i=1}^{N}$:

$$s_i = \frac{\mathbf{z}^\top \mathbf{m}_i}{\|\mathbf{z}\| \|\mathbf{m}_i\|}, \tag{1}$$

where s_i is the ith element of the similarity vector $\mathbf{s} \in \mathbb{R}^N$. Such a similarity vector $\mathbf{s}$ is then normalized with softmax over all elements, and obtain the normalized ith element as $\tilde{s}_i$. Then, the redundant items are filtered out by subtracting a threshold λ as $\hat{s}_i$:

$$\tilde{s}_i = \frac{\exp(s_i)}{\sum_{j=1}^{N} \exp(s_j)}, \tag{2}$$

$$\hat{s}_i = \frac{\max(\tilde{s}_i - \lambda, 0) \cdot \tilde{s}_i}{|\tilde{s}_i - \lambda| + \epsilon}, \tag{3}$$

where ϵ is a small positive constant to avoid division by zero. We empirically observe that the threshold $\lambda \in [1/N, 3/N]$ can lead to considerable results. This process ensures the selected items are crucial and representative. After that, we normalize the similar vector to a unit weight vector, where each item is computed as $w_i = \hat{s}_i / \sum_{j=1}^{N} \hat{s}_j$. Finally, we reintegrate $\hat{\mathbf{z}}$, which has the same dimension as $\mathbf{z}$, through the normalized weights $\mathbf{w}$ and memory $\mathcal{M}$:

$$\hat{\mathbf{z}} = \mathbf{w}^\top \mathcal{M}. \tag{4}$$

The goal of memory module is to leverage the features from highly correlated items, so as to reintegrate the encoded feature in the training process. Our memory module is constantly updated at each iteration, which can dynamically and adaptively capture the robust patterns of normal events. When it comes to the testing phase, the memory items with higher similarity can be taken to reintegrate both normal and abnormal events, in which a higher reintegration error can be achieved for the abnormal event, and a

lower reintegration error can be generated for the normal event. In summary, the memory module can further improve the ability of our MAAM-Net in distinguishing abnormal and normal events.

3.3. Training loss

3.3.1. Generative loss

We constrain the difference between the input and the generated outputs to better portrait the normal patterns with a generative loss. The loss function includes an appearance loss of the appearance branch and a motion loss of the motion branch. The appearance loss measures the similarity between the original input frame $\mathbf{I}$ and its reconstruction result $\hat{\mathbf{I}}$. Following [1,2,6,11], L_2 distance is used to compute the pixel-wise difference:

$$\mathcal{L}_{\text{app}} = \|\mathbf{I} - \hat{\mathbf{I}}\|_2. \tag{5}$$

The motion loss measures the difference (L_1 distance) of original optical flow $\mathbf{F}$ and the predicted optical flow $\hat{\mathbf{F}}$:

$$\mathcal{L}_{\text{mot}} = \|\mathbf{F} - \hat{\mathbf{F}}\|_1. \tag{6}$$

Based on the appearance loss and motion loss, the generative loss is defined as their weighted summation:

$$\mathcal{L}_{\text{gen}} = \mathcal{L}_{\text{app}} + \lambda_{\text{m}}\mathcal{L}_{\text{mot}}, \tag{7}$$

where λ_{m} controls the relative importance between the appearance and motion estimations.

3.3.2. Sparse loss and latent loss

The memory module dynamically stores a large number of normal patterns, resulting in a considerable computational burden. Moreover, since normal and abnormal events are in the same scenario and share a similar appearance feature, the generative quality of abnormal events tends to be fairly good with the complex interaction of different normal patterns, but this is not desired as we aim to differentiate the generative error between normal and abnormal events.

Therefore, it is critical to reduce the number of selected items from the memory module for reintegration, meanwhile reducing computation costs. The normal patterns in the memory module should be more sparse and general, making the generative error of abnormal events larger. Besides the generative loss, we also use the sparse loss to enforce sparsity among the selected items. Following [2], we formulate the sparse loss as:

$$\mathcal{L}_{\text{spa}} = \sum_{i=1}^{N} -w_i \cdot \log(w_i). \tag{8}$$

To further reduce the selected items of the memory module, Park et al. [10] and Chang et al. [30] manually restrict the number of normal patterns, which we argue is not appropriate. A more suitable way would be letting the memory module adaptively determine the number of normal patterns based on the normal events of a given dataset.

Since we use the normal patterns to reintegrate the encoded feature, the quality of integration could not be ideal. Inspired by the margin loss [31], we enforce a small extent of diversity for the reintegrated quality because of inadequate details. Specifically, we introduce the latent loss based on the margin between the encoded feature $\mathbf{z}$ and its reintegration $\hat{\mathbf{z}}$ as:

$$\mathcal{L}_{\text{lat}} = \max\left(\left(\left|\frac{\hat{\mathbf{z}}^{\top}\mathbf{z}}{\|\hat{\mathbf{z}}\|\,\|\mathbf{z}\|}\right| - \zeta\right), 0\right), \tag{9}$$

where ζ is a positive number controlling the tolerance of dissimilarities. By using the margin-based latent loss, the distance between normal and abnormal events in the feature space will be much larger since the abnormal events are less likely to reintegrate well with reduced items. Under the supervision of this loss, we make the following observations: (1) the number of selected items decreases significantly when applied the margin-based latent loss (Fig. 5), which represents the reduction of computation costs, we speculate the reason is that reduced items share similarly detailed information; (2) the performance is slightly improved, as shown in Table 3, while the inference speed is also improved accordingly due to the reduction of the computation costs of the memory module.

Based on the generative losses, sparse loss and the introduced latent loss, our total loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{gen}} + \alpha\mathcal{L}_{\text{spa}} + \beta\mathcal{L}_{\text{lat}}, \tag{10}$$

where α, β are hyperparameters that control the importance of different loss functions.

Table 2
Comparison with state-of-the-art methods on three benchmarks in AUC performance. Numbers in **bold** means the best results in the target dataset and the underlined numbers indicate the second-best results.

Method	Ped2	Avenue	SHTech
ConvAE [3]	0.900	0.702	0.609
AbnormalGAN [12]	0.935	–	–
ConvLSTM-AE [13]	0.881	0.770	–
TSC [1]	0.910	0.806	0.680
sRNN [1]	0.922	0.817	0.680
Zhao et al. [39]	0.912	0.771	–
Liu et al. [11]	0.954	0.851	0.728
Nguyen et al. [6]	0.962	0.869	–
MemAE [2]	0.941	0.833	0.712
Abati et al. [18]	0.954	–	0.725
RIAD [40]	0.925	0.889	–
LRCCDL [37]	0.827	0.887	–
FSCN [38]	0.928	0.855	–
NM-GAN [16]	0.963	0.886	**0.853**
Chang et al. [30]	0.962	0.860	0.733
Park et al. [10]	0.902	0.828	0.698
Ours (w/o memory)	0.947	0.895	0.681
Ours	**0.977**	**0.909**	0.713

Table 3
Comparison with state-of-the-art methods on three benchmarks in EER performance. Numbers in **bold** mean the best results in the target dataset, and the underlined numbers indicate the second-best results.

Method	Ped2	Avenue	SHTech
Sabokrou et al. [9]	13%	–	–
LRCCDL [37]	9.44%	–	–
FSCN [38]	12.5%	20.7%	–
NM-GAN [16]	**6.0%**	15.3%	**17.0%**
Ours	6.3%	**14.6%**	33.9%

3.4. Detection method

In the testing phase, an anomaly score is assigned for each frame. As presented in Fig. 3, as our method contains an appearance branch and a motion branch, we apply the detection method on each branch and fuse them to a united error map, which is obtained using L_p distance between the inputs and outputs following recent works [2,6,10]. And the error maps are used to obtain the anomaly score as well as the spatial position of abnormal events.

Notably, an abnormal event occurring within a small frame region may be missed by existing methods due to the average/summation operation over the entire frame. Even if the generative error in a distant abnormal area is high, the averaged error of the scene tends to be low, which means the generative error in a particular region is better than the whole frame for detecting anomalies. For a normal sample (Fig. 4(b)), we find that the

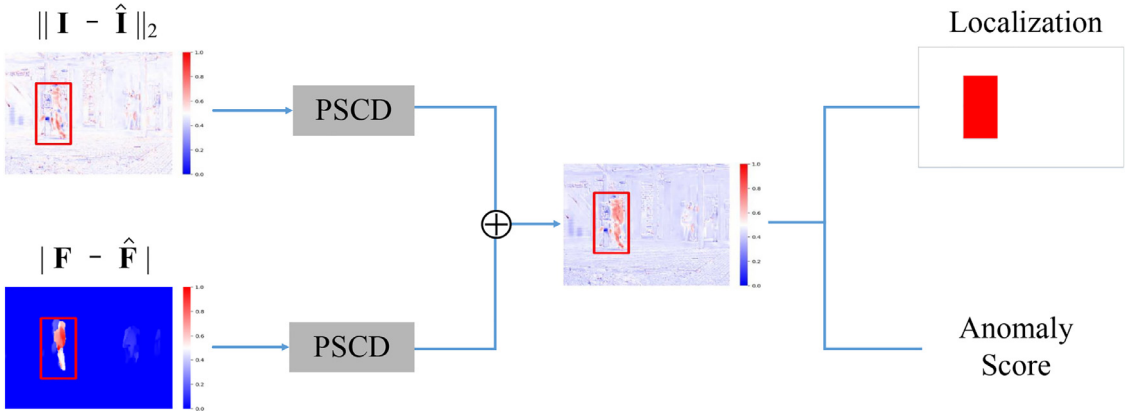
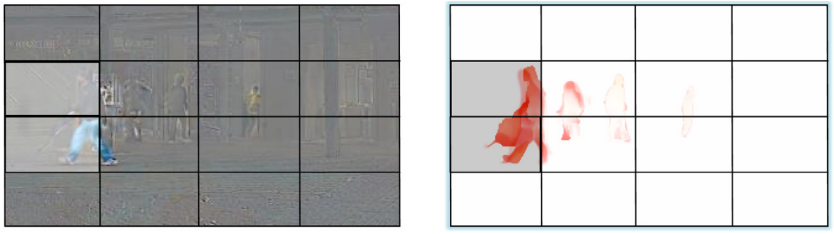
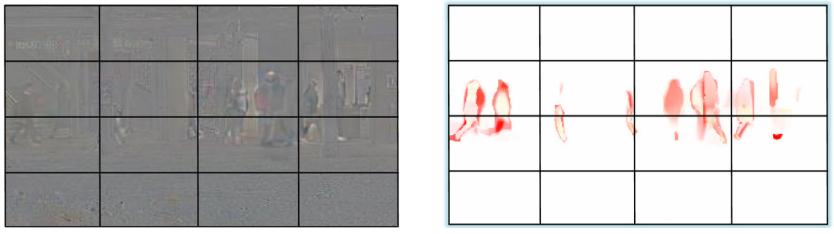


Fig. 3. Overview of the anomaly detection procedure. After obtaining the reconstructed appearance outputs and predicted motion outputs, we compute the difference between them and their original inputs, and then apply the Patch-based Stride Convolutional Detection Method (PSCD) to the difference map of appearance and motion branches. The localization map of abnormal events and the anomaly score can be calculated from the fusion map.



(a) Unnormalized abnormal event.



(b) Unnormalized normal event.

Fig. 4. The motivation of the patch-based detection method. The left column is the reconstructed error map of the appearance branch, and the right column is the predicted error map of the motion branch. The bright mask in upper left and dark mask in upper right point out the abnormal region in the patches.

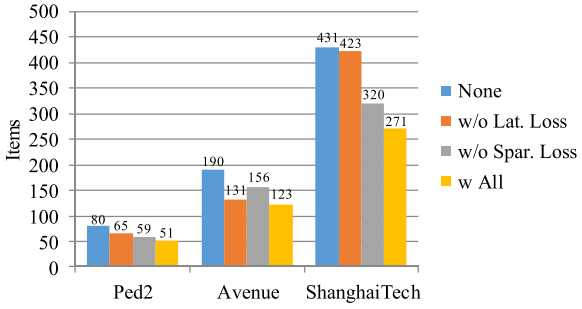


Fig. 5. Number of average support items selected from memory per sample in three different datasets. Four columns represent without sparse and latent loss, without sparse loss only, without latent loss only and with sparse and latent loss.

generative quality is also degraded, and the generative error score computed by average/summation operation may equal to the score in a tiny area of an abnormal sample (Fig. 4(a)). Furthermore, abnormal events can be perfectly detected from some well-designed patches as shown in Fig. 4(a). Thus, we propose a novel detection method considering local patches instead of the entire frame, i.e., Patch-based Stride Convolutional Detection algorithm (PSCD). The detection procedure is summarized in Algorithm 1.

The PSCD algorithm firstly initializes a filter of size $(H, W, C, 1)$ that is filled by 1, where H, W are based on the shape of a key object (e.g., a person) in the scene for a dataset and C is the input channels. Then, we apply the convolutional operation on the difference map between the input and the corresponding output of the appearance branch d_{app} and that of the motion branch d_{mot}, and generate a appearance error map e_{app} and a motion error map e_{mot}. The weighted summation between the appearance and motion error maps $e_{\text{fuse}} = \psi e_{\text{app}} + (1 - \psi) e_{\text{mot}}$ is used to obtain the final anomaly score of each patch, where ψ is a fusion hyperparameter. Next, we apply max pooling on the set of all fused error map for each frame and use the maximal patch error as the anomaly score η, Finally, we normalize the anomaly score σ to the range [0,1]:

$$\sigma = \frac{\eta - \min(\eta)}{\max(\eta) - \min(\eta)}. \tag{11}$$

A higher σ value means a higher probability of anomaly. Namely the anomaly can be detected when the score is above a pre-

Algorithm 1 The proposed PSCD anomaly detection algorithm.

Input:
The original frame $\mathbf{I}$, ground-truth flow $\mathbf{F}$, reconstruction frame $\hat{\mathbf{I}}$, reconstruction flow $\hat{\mathbf{F}}$, and h, w representing filter height and width, and the weights ψ for the fusion.
Input:
Anomaly score σ of target frame.
repeat
$d_{\text{app}} = |\mathbf{I} - \hat{\mathbf{I}}|^2$ *# Element-wise difference*
$d_{\text{mot}} = |\mathbf{F} - \hat{\mathbf{F}}|$ *# Element-wise difference*
$\text{Filter}_{\text{app}} = \text{Ones}(h, w, 3, 1)$ *# Filter filled by 1*
$\text{Filter}_{\text{mot}} = \text{Ones}(h, w, 2, 1)$ *# Filter filled by 1*
 # the parameters in the convolutional operation are feature map, filter, and stride, respectively.
$e_{\text{app}} = \text{Conv2d}(d_{\text{app}}, \text{Filter}_{\text{app}}, (H/4, W/4))$
$e_{\text{mot}} = \text{Conv2d}(d_{\text{mot}}, \text{Filter}_{\text{mot}}, (H/4, W/4))$
$e_{\text{fuse}} = \psi e_{\text{app}} + (1 - \psi) e_{\text{mot}}$ *# fusion error map*
$\eta = \max(e_{\text{fuse}})$
$\sigma = \frac{\eta - \min(\eta)}{\max(\eta) - \min(\eta)}$
Output the Anomaly score σ
until all frames are done.

defined threshold τ:

$$y = \begin{cases} 0\ , & \text{if } \sigma < \tau, \\ 1\ , & \text{otherwise.}, \end{cases} \tag{12}$$

where 0 indicates normal events while 1 indicates anomaly. Moreover, we find that the spatial position of abnormal events can be located by suppressing the non-abnormal patch.

A similar detection method is of Nguyen et al. [6], while the major differences between our proposed PSCD and them lie in two aspects: (1) the patch size is not designed as the same size in all datasets, since we regularize the patch size as the shape of people shown in the datasets; (2) the calculation of the difference map is similar with the loss functions while [6] use the mean square error for appearance branch and motion branch, and the fusion mechanism of two branches is the weighted summation without the supervision of training samples while [6] is not.

4. Experiments and discussions

In this section, we present the implementation details of the proposed MAAM-Net and compare it with the state-of-the-art methods on three benchmark datasets, including UCSD Ped2 [32], CUHK Avenue [33], and ShanghaiTech [1]. Extensive ablation studies are then performed to validate the contribution of each component of the MAAM-Net.

4.1. Datasets

We evaluate our proposed method on three benchmarks: UCSD Ped2, CUHK Avenue and ShanghaiTech, which are introduced in the following paragraphs.

The UCSD Ped2 dataset [32] contains 16 training videos and 12 testing videos with 12 kinds of abnormal events. The frame is in grey scale. There are 2550 samples for training and 2010 samples for testing. Abnormal events include vehicles such as bicycles, cars and skateboards. What's challenging, a scene may contain multiple anomalies. The camera is fixed and the size of people in the scenes is almost the same.

The CUHK Avenue dataset [33] contains 16 training videos and 21 testing ones with a total of 47 kinds of abnormal events, including throwing objects, loitering, running, dancing, etc. The total numbers of frames for training and testing are 15,328 and 15,324, respectively. The camera is also fixed but the size of people may differ because of the distance to cameras.

The ShanghaiTech dataset [1] is so far the biggest and most challenging dataset. It contains 330 training videos of 274,515 frames and 107 testing videos of 42,883 frames with 130 kinds of abnormal events. The main differences between it and the aforementioned two datasets are that it contains 13 different scenes, and the abnormal events include running, bicycles, vehicles, fighting, etc. Due to the different scenes, the size of objects varies from scene to scene.

The above three datasets contain training and testing sets receptively. We directly use the training set for training our MAAM-Net and testing set for evaluation on each experiment.

4.2. Evaluation metrics

Following previous works [6,10,30,34], we also evaluate our proposed MAAM-Net by the Area Under Curve (AUC), which is obtained by calculating the area under the Receiver Operation Characteristic (ROC) curve. We further report the Equal Error Rate (EER), which is the point on the ROC curve that corresponds to having an equal probability of miss-classifying a positive or negative sample, to validate the effectiveness of the MAAM-Net. Since few papers have been published with the EER score, we present it as a trigger for future work comparison.

4.3. Implementation details

The input size of the frame is set to 180×320 for Avenue and ShanghaiTech datasets. Due to the frame size in Ped2 dataset is small, we directly use the frames of their original size to train our model. The corresponding optical flow [29] is also resized to the same resolution with bilinear interpolation. The flow visualization is obtained by using the official code of Flownet2 [29]. All experiments are conducted on a single NVIDIA RTX 2080 Ti GPUs.

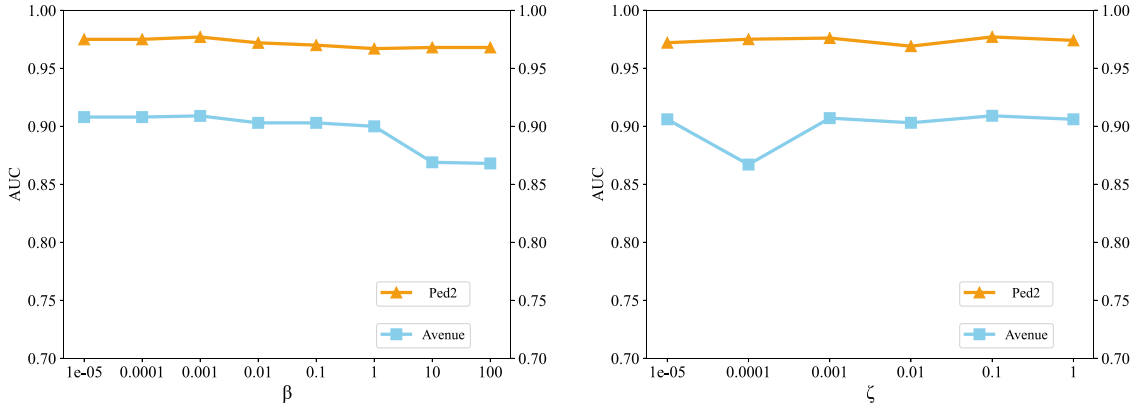
Our proposed MAAM-Net is implemented with Tensorflow (version 1.13.1) [35] in Ubuntu (version 16.04.7). We use the Adam optimizer [36] with a fixed learning rate (0.00002, 0.0005, and 0.00001 on Ped2, Avenue, and ShanghaiTech, respectively) during the whole training process. In practice, the memory capacity is set to 500 for Ped2 and 2000 for other two datasets, λ is $1/N$ (N is memory capacity), and the margin is 0.001. We empirically set λ_{m}, α, β, and ζ as 2, 0.0003, 0.001, and 0.1 on all datasets following [2,6], respectively. The training epoch is set to 50. The patch size is set to the average size of persons shown in the scenes.

4.4. Comparison with state-of-the-arts

We present the AUC and EER performance comparison with state-of-the-art methods on the Ped2 [32], Avenue [33], and ShanghaiTech [1] datasets in Tables 2 and 3, respectively. Among the compared methods, we re-implement the performance of Nguyen et al. [6] and Sabokrou et al. [9], and directly report the performance of the remaining methods.

4.4.1. Ped2

Our MAAM-Net achieves an AUC of 0.977 on the Ped2 dataset, which outperforms all previous methods. Compared with the baseline MemAE [2], our method outperforms it by a margin of 3.6% in AUC, which demonstrate the efficacy of the introduction of the motion branch. Moreover, our method can further obtain a 1.5% gain in AUC when competing with the appearance-motion model that lacks of the memory module [6], which demonstrate the necessity and effectiveness of the combination of them. Our MAAM-Net also achieves a considerable EER performance of 6.3%, which outperforms recent methods [9,37,38] and is 0.3% lower than NM-GAN [16] using an additional discriminator.



(a) Weight (b) Margin

Fig. 6. Hyperparameter sensitivity analysis on the weight parameter β and the margin parameter ζ. The Ped2 dataset uses the primary axis and the Avenue dataset uses the deputy axis.

4.4.2. Avenue

Compared with the Ped2 dataset, Avenue is a larger and more challenging dataset due to more anomaly scenes. On this dataset, our MAAM-Net still outperforms all competing methods in AUC performance and improves the AUC to a new stage. The performance in EER is competitive as well. Compared with Park et al. [10], which leverages manual intervention of memory update, our method achieves an 8.1% higher AUC and exceeds our baseline MemAE [2] by a large margin of 7.6% in AUC. Even for MAAM-Net without the memory module, our method still has comparable performance, validating the superior capability of our proposed MAAM-Net. Moreover, our MAAM-Net outperforms the recent method FSCN [38] both in AUC and EER, and is 2.3% higher in AUC and is 0.7% higher in EER compared to the recent method NM-GAN [16].

4.4.3. ShanghaiTech

We achieve competitive performance with recent methods [1–3,10] in AUC. Notably, we outperform the recent methods that leverage memory modules [2,10]. We note several methods [11,16,18,30] achieve higher AUC performance than ours on this dataset, and we speculate the reason is that they use additional training clues. Specifically, they can benefit from multi-frame input [11,18,30] and/or a much longer training schedule [16], while our model only needs single-frame input and requires fewer epochs to achieve considerable performance in contrast, which also leads to a high ratio of false detection. The other reason might be that our detection method is based on patches, and the patch size is based on the average size of persons in the dataset. Frames in ShanghaiTech exhibit large scene variances than those in the Ped2 and Avenue datasets, which leads to the performance degradation of our PSCD algorithm. Our EER performance is relatively weak compared with the recent method NM-GAN [16], and we speculate the reasons are as follows. The first one is NM-GAN [16] heavily relies on a pre-defined normal distribution to produce negative samples in the training phase, while the normal events are enough for our MAAM-Net. The second one is the input of NM-GAN [16] is not the whole frame but the overlapped patch sets of training samples. Although the two additional works make a better EER of NM-GAN [16], they need some preliminary work, leading to a massive repetitive computation and slowing the inference speed.

To summarize, with the help of the memory module and the novel Patch-based Stride Convolutional Detection (PSCD) algorithm, our method outperforms all competing methods on the Ped2 and Avenue datasets in AUC, and achieves competitive AUC performance on the ShanghaiTech dataset with recent methods [1–3,10,11,18,30]. Besides, our method achieves comparable EER performance with other methods [9,37,38] on Ped2 and Avenue datasets. It is close to NM-GAN [16] on the Ped2 and Avenue datasets but poor than NM-GAN on the ShanghaiTech dataset in EER. Moreover, even without the memory module, our model still achieves a comparable performance, verifying the intuition of introducing motion features into MAAM-Net.

Table 4
Comparison with various methods on model complexity and inference speed on three benchmarks. "M" means millions, and "FPS" denotes Frames Per Second. Numbers in **bold** represent the best results.

Methods	Parameters (M)	FPS		
		Ped2	Avenue	SHTech
Liu et al. [11]	7.7	32.76	20.97	16.17
MemAE [2]	**6.5**	59.14	–	–
AMMC-Net [19]	25.1	37.03	44.11	40.18
Ours	8.5 (9.22)	**63.6**	**78.38**	**79.76**

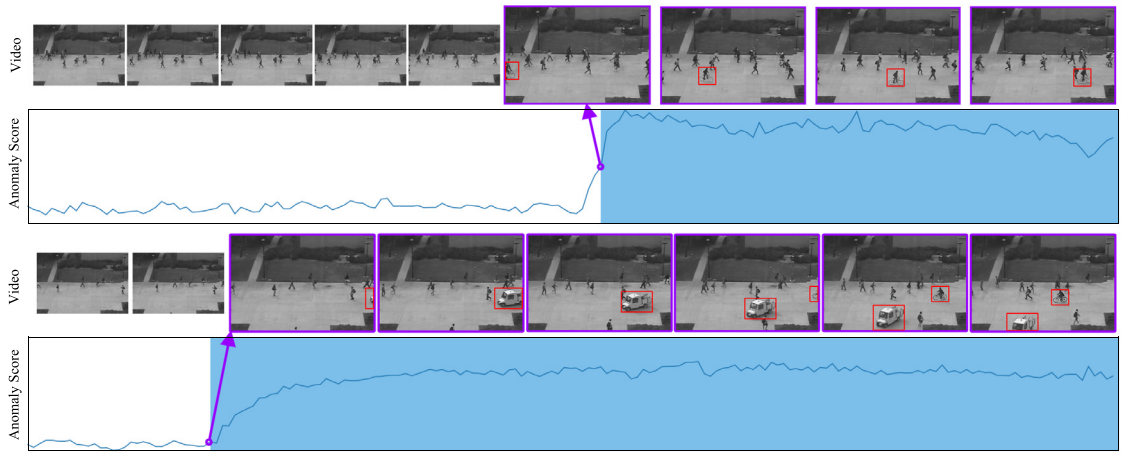
4.5. Model complexity and inference speed

We present the model complexity and inference speed comparison with other methods [2,11,19] on the Ped2, Avenue, and ShanghaiTech datasets in Table 4. The inference speeds of different methods are collected by running the official implements on a single Nvidia RTX-2080Ti GPU. Here, we present two values of the model parameter of our method, which means 8.5 M on Ped2 and 9.22 M on Avenue and ShanghaiTech. The reason for the different model parameters is videos in Avenue and ShanghaiTech are more complex than Ped2, and thus we use bigger memory capacity for them. Although the parameter size of MemAE [2] is smaller than that of ours, the large selected items used in MemAE make the inference procedure more time-consuming. Apart from the model parameter, our model is faster than all competing methods as it can infer 63.6 FPS on Ped2, 78.38 FPS on Avenue, and 79.76 FPS on ShanghaiTech, verifying the intuition that improving the feature generalization of the memory module is efficient to accelerate the inference speed.

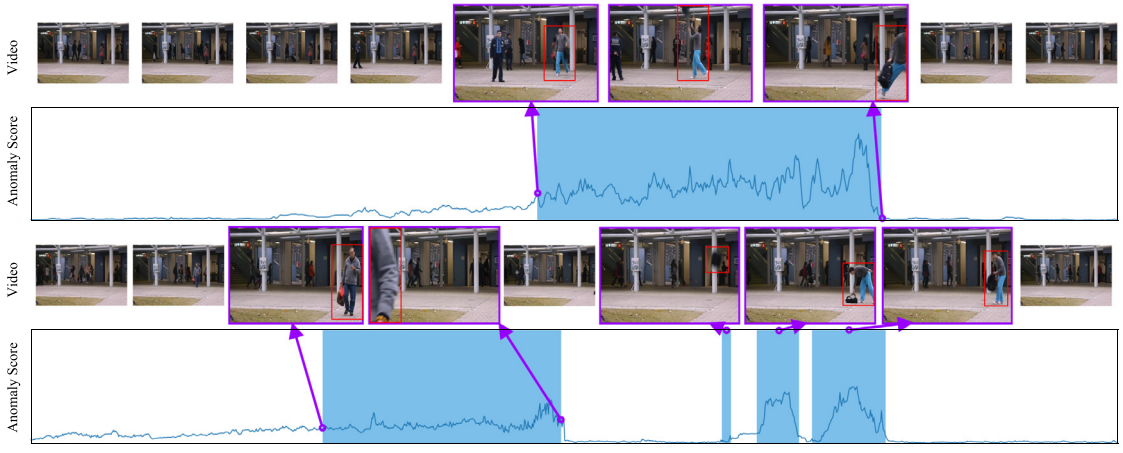
4.6. Ablation study

4.6.1. Appearance and motion branches

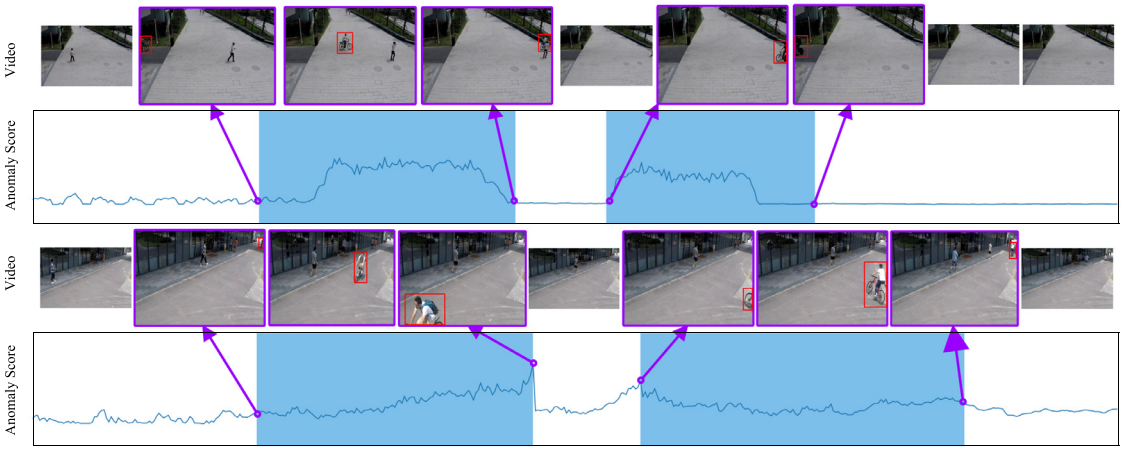
In our method, we use an appearance branch to reconstruct the input frame and learn spatial features, and use a motion



(a) Detection examples from the Ped2 dataset.



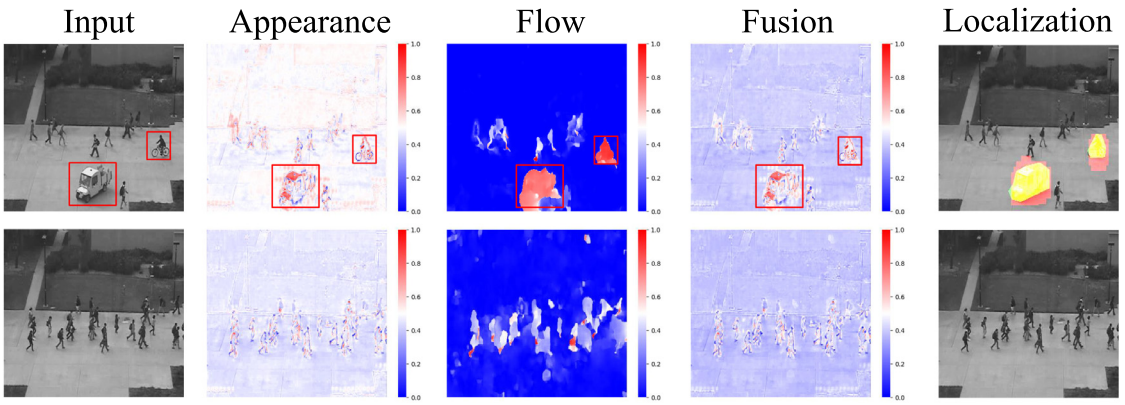
(b) Detection examples from the Avenue dataset.



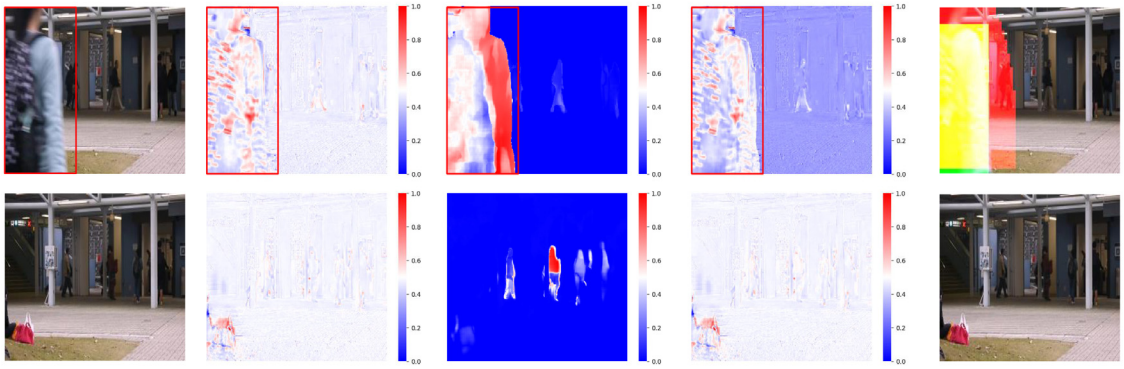
(c) Detection examples from the ShanghaiTech dataset.

Fig. 7. Quantitative results on three datasets. Each dataset contains two examples and each example have two rows, the first is the video sequence, the abnormal events are enclosed by red boxes and the abnormal frames are larger than normal events, the second row is the corresponding anomaly score for each frame, higher anomaly score represents higher probability of abnormalities. The light blue background area is the ground-truth anomaly range in temporal sequence. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

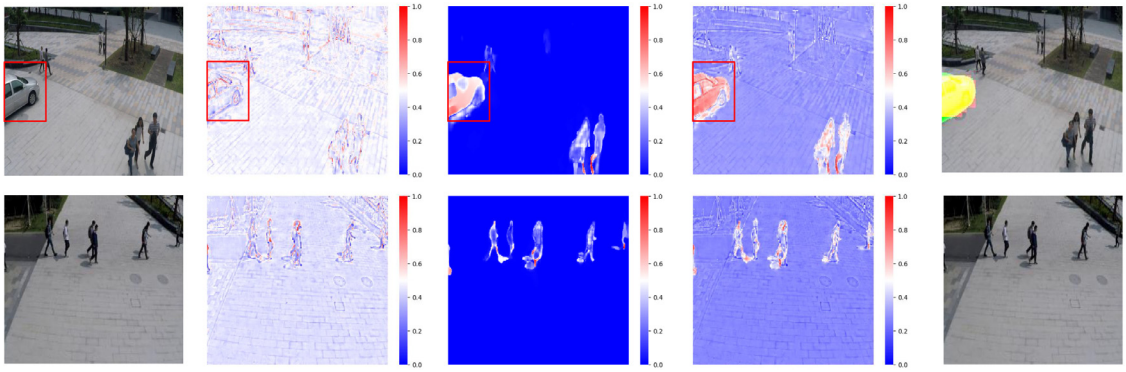
branch to predict the optical flow and capture the temporal correlation. To validate the necessity of the two branches, we conduct ablation experiments on Ped2, Avenue, and ShanghaiTech datasets. The last row of Table 4 summarizes the performance of using different branches on the three datasets. The results suggest the appearance-motion architecture largely improves the detection performance. We argue the motion feature plays a more important role than the appearance feature since abnormal events are often positively correlated with unusual speed of object movement, and the visualization results are presented in Section 4.7. Therefore, the



(a) Testing samples from the Ped2 dataset.



(b) Testing samples from the Avenue dataset.



(c) Testing samples from the ShanghaiTech dataset.

Fig. 8. Detailed visualization samples. Each row contains original inputs, appearance differences, motion differences, fusion differences and anomaly localization respectively. In the last column, the red, green, and yellow masks represent the predicted abnormal area, ground-truth abnormal area and the overlap area of them. The first two rows are from the Ped2 dataset, the next two rows are from the Avenue dataset and the last two rows are from the ShanghaiTech dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

motion branch can better model the anomaly, and thus can achieve higher performance than the appearance branch. The combination of them further boosts the performance.

4.6.2. Different losses

In our proposed MAAM-Net, we introduce a sparse loss $\mathcal{L}_{\text{spa}}$ and a latent loss $\mathcal{L}_{\text{lat}}$ to reduce the selected items of the memory module for feature reintegration. Specifically, the sparse loss encourages the selected items to be sparse and the latent loss forces the selected items to be critical, and they together improve the generalization of the memory module. Due to the natural relationship between normal events and abnormal events, i.e., anomalies usually exhibit anomalous shapes and speeds, it has no effect for normal events to reintegrate when we slightly drop the reintegration quality of normal events, while it increases the difficulty to reintegrate abnormal events. Thus it validates the effectiveness of the latent loss and sparse loss.

Table 5 lists the performance of all combinations of losses with different branch mixtures. By using the sparse loss and the latent loss, the performance outperforms all variants that use the appearance-motion architecture. Besides, in single-branch cases, the combination of two loss terms may underperform a single loss, and we speculate the reason is that the two loss functions both aim to reduce the selected items for reintegration, which restrict

Table 5
Ablation Study on loss and branch combinations. The performance is measured by AUC. Numbers in **bold** represents the best results. The tick in $\mathcal{L}_{spa}$ and $\mathcal{L}_{lat}$ means whether using the target loss.

$\mathcal{L}_{spa}$	$\mathcal{L}_{lat}$	Ped2			Avenue			SHTech		
		App.	Mot.	App. + Mot.	App.	Mot.	App. + Mot.	App.	Mot.	App. + Mot.
		0.826	0.972	0.974	0.782	0.828	0.891	0.511	**0.713**	0.711
✓		**0.855**	**0.972**	0.974	**0.784**	0.823	0.899	0.509	0.712	0.707
	✓	0.830	0.972	0.975	0.779	**0.834**	0.889	0.507	0.711	0.709
✓	✓	0.778	0.972	**0.977**	0.773	0.820	**0.909**	**0.539**	0.712	**0.713**

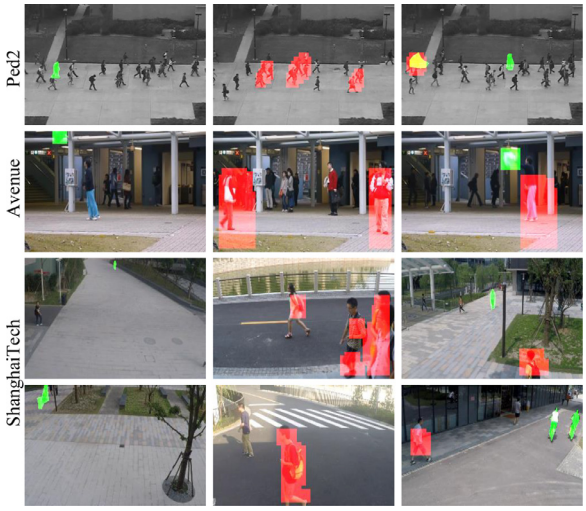


Fig. 9. Failure cases. The red, green, and yellow masks represent the predicted abnormal area, ground-truth abnormal area, and the overlap area of them. The first two rows are from Ped2 and Avenue dataset respectively, and the last two rows are from ShanghaiTech dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the single-branch model to learn rich features. Without the interaction of both branches, the performance with latent loss in the single branch model is weak compared with other combinations of loss functions, which indicates that the latent loss heavily relies on the representational ability of features.

4.6.3. Computation cost

In order to verify the influence that the new loss function on the decline of selected items, we also present the average number of the reintegration items for each sample among three datasets in Fig. 5. It indicates that the new latent loss largely reduces the number of selected items. With the latent loss, the proportion of selected items drops around 25%, 36%, and 37% on Ped2, Avenue, and ShanghaiTech datasets respectively, which validates the statement that the introduced latent loss is able to reduce the computation costs and further improve the feature generalization. Furthermore, with the sparse loss, the number of the selected items proceeds to be reduced.

4.6.4. Hyperparameter sensitivity

There are two important hyperparameters in our proposed MAAM-Net, i.e., the weight parameter β and the margin parameter ζ. The sensitivity analysis of them are presented in Fig. 6.

In particular, β controls the importance of feature sparsity, and further influences the feature generalization of the memory module. Figure 6(a) plots the AUC performances under different values of β. We can observe that the performance of Avenue has an obvious drop when the weight is larger than 1. While the performance is not sensitive to the variation of the weight parameter when it is smaller than 1. Such phenomenon validates the necessity of the introduction of our latent loss.

What's more, ζ controls the tolerance of the model for the generative quality. A large ζ makes the memory module select the scarce yet crucial items for reintegration, but causes a quality drop. Figure 6(b) plots the AUC performances of different ζ on Ped2 and Avenue datasets. As the cosine similarity is smaller than 1, we set different ζ in the range of [0,1]. We can see that our model is not sensitive to the variation of the margin parameter ζ.

4.7. Qualitative evaluation

4.7.1. Qualitative analysis on testing videos

Figure 7 presents six qualitative results on the Ped2, Avenue, and ShanghaiTech datasets, respectively. We showcase two video examples for each dataset. The abnormal event is enclosed by red boxes and the frame it belongs to is larger than normal frames. We can see that the abnormal events usually correspond to the anomaly score raise. With the increase of the dataset complexity (Ped2 < Avenue < ShanghaiTech), it is much more difficult to isolate the normal and abnormal events.

Specifically, in Fig. 7(a), we can clearly observe that MAAM-Net can successfully isolate the normal and abnormal events. When the bicycle or the car appears in the scene, the anomaly score raises significantly. The abnormal labels perfectly correspond to the predicted results, which shows the efficacy of our proposed MAAM-Net. In Fig. 7(b), the boundary between the normal and abnormal events is also clearly enough. The emerge of abnormal events (A man throwing a bag and a man walking in the wrong direction) incur high anomaly scores. Compared with the Ped2 dataset, Avenue dataset is much more complicated, and thus the curve of anomaly scores appear to fluctuate, which makes it a little difficult to choose a suitable threshold. In Fig. 7(c), the curve becomes more fluctuant due to the scene variances in ShanghaiTech dataset. The results show that the anomalies (the bicycles) detected by our method are well located in the ground-truth temporal sequence.

4.7.2. Qualitative analysis on testing frames

Figure 8 plots some testing samples. We can see that the fusion results of appearance and motion branches surely improve the local response and widen the difference between the normal and abnormal areas of each sample, and meanwhile suppress the response of the normal samples. As illustrated in the last column, the red, green, and yellow masks represent the predicted abnormal area, ground-truth abnormal area, and the overlap area of them, we can see that our method perfectly locates the spatial location of the abnormal events, which demonstrates that the MAAM-Net has the ability to locate the specific anomaly region without additional bounding box annotations.

4.7.3. Failure cases

Under the unsupervised learning setting of anomaly detection, the model is prone to misclassify hard positive samples due to the lack of abnormal events during training. Figure 9 shows several failure examples, where some testing samples are misclassi-

fied due to the complexity of human behaviors and the distance from the surveillance camera. However, to clearly distinguish the hard positive examples from the true negative examples may require stronger supervision.

5. Conclusion

In this paper, we propose a Memory-Augmented Appearance-Motion Network (MAAM-Net) for video anomaly detection, which benefits from a novel end-to-end memory-augmented network for learning the appearance and motion feature of a given input frame, a well-designed margin-based latent loss, and a pointed Patch-based Stride Convolutional Detection (PSCD) algorithm. The memory module is creatively embedded into the appearance-motion base network, which can differentiate the normal and abnormal events by using the feature representation of the former to reintegrate the latter. The margin-based latent loss forces the memory module to select a sparse set of critical items for reintegration and further reduces the computation costs. The PSCD algorithm focuses on the patch-level rather than the frame-level response of the error map, which explicitly utilizes the characteristics of the abnormal events and boosts detection accuracy. Experiments on three benchmarks demonstrate the efficacy and efficiency of the proposed MAAM-Net compared to the state-of-the-art methods.

6. Limitation and future work

Our proposed PSCD algorithm has an important value for improving anomaly detection accuracy. It decreases the potential false alarms caused by the long-range abnormal region. However, it performs weakly in the multi-scene anomaly dataset, since choosing a suitable patch size is difficult and thus degrades the performance. Therefore, we plan to adjust the mechanism of patch choice to adapt to the needs of multi-scene anomaly detection in future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

No data was used for the research described in the article.

Acknowledgments

This work was supported partly by National Key R&D Program of China under Grant 2018AAA0101400, NSFC under Grants 62088102, 61976171, and 62106192, China Postdoctoral Science Foundation under Grant 2020M683490, Natural Science Foundation of Shaanxi Province under Grant 2021JQ-054, and Fundamental Research Funds for the Central Universities under Grant XTR042021005.

References

[1] W. Luo, W. Liu, S. Gao, A revisit of sparse coding based anomaly detection in stacked RNN framework, in: Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 341–349.
[2] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, H.A. van den, Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1705–1714.
[3] M. Hasan, J. Choi, J. Neumann, A.K. Roy-Chowdhury, L.S. Davis, Learning temporal regularity in video sequences, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 733–742.
[4] Z. Liu, L. Wang, Q. Zhang, W. Tang, J. Yuan, N. Zheng, G. Hua, ACSNet: action–context separation network for weakly supervised temporal action localization, in: Proc. AAAI Conf. Artif. Intell., 2021, pp. 2233–2241.
[5] Y. Zhai, L. Wang, W. Tang, Q. Zhang, J. Yuan, G. Hua, Two-stream consensus networks for weakly-supervised temporal action localization, in: Proc. Eur. Conf. Comput. Vis., 2020, pp. 37–54.
[6] T.-N. Nguyen, J. Meunier, Anomaly detection in video sequence with appearance-motion correspondence, in: Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 1273–1283.
[7] R.T. Ionescu, F.S. Khan, M.-I. Georgescu, L. Shao, Object-centric auto-encoders and dummy anomalies for abnormal event detection in video, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 7842–7851.
[8] P. Perera, R. NFallapati, B. Xiang, OCGAN: one-class novelty detection using GANs with constrained latent representations, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2898–2906.
[9] M. Sabokrou, M. Khalooei, M. Fathy, E. Adeli, Adversarially learned one-class classifier for novelty detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3379–3388.
[10] H. Park, J. Noh, B. Ham, Learning memory-guided normality for anomaly detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 14372–14381.
[11] W. Liu, W. Luo, D. Lian, S. Gao, Future frame prediction for anomaly detection - a new baseline, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 6536–6545.
[12] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, N. Sebe, Abnormal event detection in videos using generative adversarial nets, in: Proc. IEEE Int. Conf. Image Process., 2017, pp. 1577–1581.
[13] W. Luo, W. Liu, S. Gao, Remembering history with convolutional LSTM for anomaly detection, in: Proc. Int. Conf. Multimedia Expo, 2017, pp. 439–444.
[14] S. Akcay, A. Atapour-Abarghouei, T.P. Breckon, GANomaly: semi-supervised anomaly detection via adversarial training, in: Proc. Asian Conf. Comput. Vis., 2019, pp. 622–637.
[15] K. Doshi, Y. Yilmaz, Online anomaly detection in surveillance videos with asymptotic bound on false alarm rate, Pattern Recognit. 114 (2021) 107865.
[16] D. Chen, L. Yue, X. Chang, M. Xu, T. Jia, NM-GAN: noise-modulated generative adversarial network for video anomaly detection, Pattern Recognit. 116 (2021) 107969.
[17] A. Acsintoae, A. Florescu, M.-I. Georgescu, T. Mare, P. Sumedrea, R.T. Ionescu, F.S. Khan, M. Shah, Ubnormal: new benchmark for supervised open-set video anomaly detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 20143–20153.
[18] D. Abati, A. Porrello, S. Calderara, R. Cucchiara, Latent space autoregression for novelty detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 481–490.
[19] R. Cai, H. Zhang, W. Liu, S. Gao, Z. Hao, Appearance-motion memory consistency network for video anomaly detection, in: Proc. AAAI Conf. Artif. Intell., 2021, pp. 938–946.
[20] S. Wisdom, T. Powers, J. Pitton, L. Atlas, Interpretable recurrent neural networks using sequential sparse recovery, arXiv preprint arXiv:1503.01007(2016).
[21] G. Yu, S. Wang, Z. Cai, X. Liu, C. Xu, C. Wu, Deep anomaly discovery from unlabeled videos via normality advantage and self-paced refinement, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2022, pp. 13987–13998.
[22] G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, D. Huang, Video anomaly detection by solving decoupled spatio-temporal jigsaw puzzles, in: Proc. Eur. Conf. Comput. Vis., 2022, pp. 494–511.
[23] Z. Liu, Y. Nie, C. Long, Q. Zhang, G. Li, A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction, in: Proc. IEEE Int. Conf. Comput. Vis., 2021, pp. 13588–13597.
[24] J. Weston, S. Chopra, A. Bordes, Memory networks, in: Proc. Int. Conf. Learn. Rep., 2015, pp. 1130–1150.
[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[26] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, in: Proc. Int. Conf. Mach. Learn., 2015, pp. 448–456.
[27] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proc. Int. Conf. Mach. Learn., 2010, pp. 807–814.
[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (56) (2014) 1929–1958.
[29] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, FlowNet 2.0: evolution of optical flow estimation with deep networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2462–2470.
[30] Y. Chang, Z. Tu, W. Xie, J. Yuan, Clustering driven deep autoencoder for video anomaly detection, in: Proc. Eur. Conf. Comput. Vis., 2020, pp. 329–345.
[31] Y. Lin, A note on margin-based loss functions in classification, Stat. Probab. Lett. 68 (1) (2004) 73–82.
[32] A.B. Chan, N. Vasconcelos, Modeling, clustering, and segmenting video with mixtures of dynamic textures, IEEE Trans. Pattern Anal. Mach. Intell. 30 (5) (2008) 909–926.
[33] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 FPS in MATLAB, in: Proc. IEEE Int. Conf. Comput. Vis., 2013, pp. 2720–2727.
[34] G. Pang, C. Yan, C. Shen, H.A. van den, X. Bai, Self-trained deep ordinal regression for end-to-end video anomaly detection, in: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 12173–12182.
[35] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A.

Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv: 1603.04467(2016).

[36] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: Proc. Int. Conf. Learn. Rep., 2015.

[37] A. Li, Z. Miao, Y. Cen, X.-P. Zhang, L. Zhang, S. Chen, Abnormal event detection in surveillance videos based on low-rank and compact coefficient dictionary learning, Pattern Recognit. 108 (2020) 107355.

[38] P. Wu, J. Liu, M. Li, Y. Sun, F. Shen, Fast sparse coding networks for anomaly detection in videos, Pattern Recognit. 107 (2020) 107515.

[39] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, X.-S. Hua, Spatio-temporal autoencoder for video anomaly detection, in: Proc. ACM Int. Conf. Multimedia, 2017, pp. 1933–1941.

[40] V. Zavrtanik, M. Kristan, D. Skočaj, Reconstruction by inpainting for visual anomaly detection, Pattern Recognit. 112 (2020) 107706.

Le Wang received the B.S. and Ph.D. degrees in Control Science and Engineering from Xi'an Jiaotong University, Xi'an, China, in 2008 and 2014, respectively. From 2013 to 2014, he was a visiting Ph.D. student with Stevens Institute of Technology, Hoboken, New Jersey, USA. From 2016 to 2017, he is a visiting scholar with Northwestern University, Evanston, Illinois, USA. He is currently an Associate Professor with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision, pattern recognition, and machine learning. He is an area chair of CVPR' 2022, and a senior program committee member of AAAI' 2022. He holds 7 China patents and has 16 more China patents pending. He is the author of more than 60 peer reviewed publications in prestigious international journals and conferences. He is a senior member of the IEEE.

Junwen Tian received the B.S. degree in Software Engineering from Taiyuan University of Technology, Taiyuan, China, in 2018. He is currently pursuing the M.S. degree in Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University. His research interests include computer vision and machine learning.

Sanping Zhou received the Ph.D. degree in control science and engineering from Xian Jiaotong University, Xian, China, in 2020. From 2018 to 2019, he was a Visiting Ph.D. Student with the Robotics Institute, Carnegie Mellon University. He is currently an Assistant Professor with the Institute of Artificial Intelligence and Robotics, Xian Jiaotong University. His research interests include machine learning, deep learning, and computer vision, with a focus on person re-identification, salient object detection, medical image segmentation, image classification, and visual tracking.

Haoyue Shi received the B.S. degree in Software Engineering from Northwest A&F University, Yangling, China in 2019. She is currently a Ph.D. student with the Institute of Artificial Intelligence and Robotics of Xi'an Jiaotong University, Xi'an, China. Her research interests lie in computer vision and machine learning.

Gang Hua was enrolled in the Special Class for the Gifted Young of Xi'an Jiaotong University (XJTU), Xi'an, China, in 1994 and received the B.S. degree in Automatic Control Engineering from XJTU in 1999. He received the M.S. degree in Control Science and Engineering in 2002 from XJTU, and the Ph.D. degree in Electrical Engineering and Computer Science at Northwestern University, Evanston, Illinois, USA, in 2006. He is currently the Vice President and Chief Scientist of Wormpex AI Research. Before that, he served in various roles at Microsoft (2015–18) as the Science/Technical Adviser to the CVP of the Computer Vision Group, Director of Computer Vision Science Team in Redmond and Taipei ATL, and Principal Researcher/Research Manager at Microsoft Research. He was an Associate Professor at Stevens Institute of Technology (2011–15). During 2014–15, he took an on leave and worked on the Amazon-Go project. He was an Visiting Researcher (2011–14) and a Research Staff Member (2010–11) at IBM Research T. J. Watson Center, a Senior Researcher (2009–10) at Nokia Research Center Hollywood, and a Scientist (2006–09) at Microsoft Live labs Research. He is an associate editor of TIP, TCSVT, CVIU, IEEE Multimedia, TVCJ and MVA. He also served as the Lead Guest Editor on two special issues in TPAMI and IJCV, respectively. He is a general chair of ICCV'2025. He is a program chair of CVPR' 2019&2022. He is an area chair of CVPR' 2015&2017, ICCV' 2011&2017, ICIP' 2012&2013&2016, ICASSP' 2012&2013, and ACM MM 2011&2012&2015&2017. He is the author of more than 200 peer reviewed publications in prestigious international journals and conferences. He holds 19 US patents and has 15 more US patents pending. He is the recipient of the 2015 IAPR Young Biometrics Investigator Award for his contribution on Unconstrained Face Recognition from Images and Videos, and a recipient of the 2013 Google Research Faculty Award. He is an IEEE Fellow, an IAPR Fellow, and an ACM Distinguished Scientist.