Pattern Recognition 122 (2022) 108213

Contents lists available at ScienceDirect

Pattern Recognition

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

Video anomaly detection with spatio-temporal dissociation

Yunpeng Chang[a], Zhigang Tu[a,*], Wei Xie[b], Bin Luo[a], Shifu Zhang[c], Haigang Sui[a], Junsong Yuan[d]

[a] The State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, Hubei 430079, China
[b] The School of Computer, Central China Normal University, LuoyuRoad 152, Wuhan, Hubei, China
[c] Shenzhen Infinova Company Ltd., Shenzhen, Guangdong 518100, China
[d] The Computer Science and Engineering department, State University of New York at Buffalo, NY 14260-2500, USA

ARTICLE INFO

Article history:
Received 29 December 2020
Revised 23 June 2021
Accepted 27 July 2021
Available online 5 August 2021

Keywords:
Video anomaly detection
Spatio-temporal dissociation
Simulate motion of optical flow
Deep K-means cluster

ABSTRACT

Anomaly detection in videos remains a challenging task due to the ambiguous definition of anomaly and the complexity of visual scenes from real video data. Different from the previous work which utilizes reconstruction or prediction as an auxiliary task to learn the temporal regularity, in this work, we explore a novel convolution autoencoder architecture that can dissociate the spatio-temporal representation to separately capture the spatial and the temporal information, since abnormal events are usually different from the normality in appearance and/or motion behavior. Specifically, the spatial autoencoder models the normality on the appearance feature space by learning to reconstruct the input of the first individual frame (FIF), while the temporal part takes the first four consecutive frames as the input and the RGB difference as the output to simulate the motion of optical flow in an efficient way. The abnormal events, which are irregular in appearance or in motion behavior, lead to a large reconstruction error. To improve detection performance on fast moving outliers, we exploit a variance-based attention module and insert it into the motion autoencoder to highlight large movement areas. In addition, we propose a deep K-means cluster strategy to force the spatial and the motion encoder to extract a compact representation. Extensive experiments on some publicly available datasets have demonstrated the effectiveness of our method which achieves the state-of-the-art performance. The code is publicly released at the link[1].

1. Introduction

Anomaly detection in videos refers to the identification of events that deviate from the expected behavior [1,2], which is an important task in video analytics and plays a crucial role in video surveillance. However, video anomaly detection is an extremely challenging task due to the following reasons: first, realistic video data is complex, and some anomaly data points may lie close to the boundary of normal regions, e.g. skateboarders and walking people are similar in appearance, but skateboarders are abnormal objects which are prohibited on the pedestrian footpath. Second, the labeled training data for anomaly detection is limited. Although the normal patterns are usually relatively easy to collect, while the abnormal samples are few and costly to acquire. Consequently, in some cases, anomaly detection methods [3,4] only train their models on the normal data to learn the regularity with an unsupervised setting, and determine which instance is deviated from the normal patterns.

Recently, many deep learning-based methods [1,5–7] have been proposed to handle the problem of limited labeled data by modeling the normal pattern. Most of these methods learn an autoencoder or U-Net to reconstruct normal events or predict future frames to capture the normality behind the video sequences. The reconstruction based anomaly detection methods [1] take the hand-crafted feature (e.g. low-level trajectory features) or directly use video frames as the input, and extract high-level feature representation to model the normality, where the temporal regularity of the normal events can be learned by minimizing the reconstruction error. Since these models only learn the patterns within the normal training set, the abnormal patterns will lead to larger reconstruction error. Therefore, abnormal events can be distinguished by their reconstruction quality. In [3], it is argued that because of the high capacity of the deep neural networks, the reconstruction error of abnormal events is not necessarily larger than that of the normal events, thus [3] proposed a prediction-based anomaly detection method, which predicts the next future frame from the previous consecutive frames with an U-Net architecture,

* Corresponding author.
E-mail address: tuzhigang@whu.edu.cn (Z. Tu).

[1] https://github.com/ChangYunPeng/VideoAnomalyDetection

https://doi.org/10.1016/j.patcog.2021.108213
0031-3203/

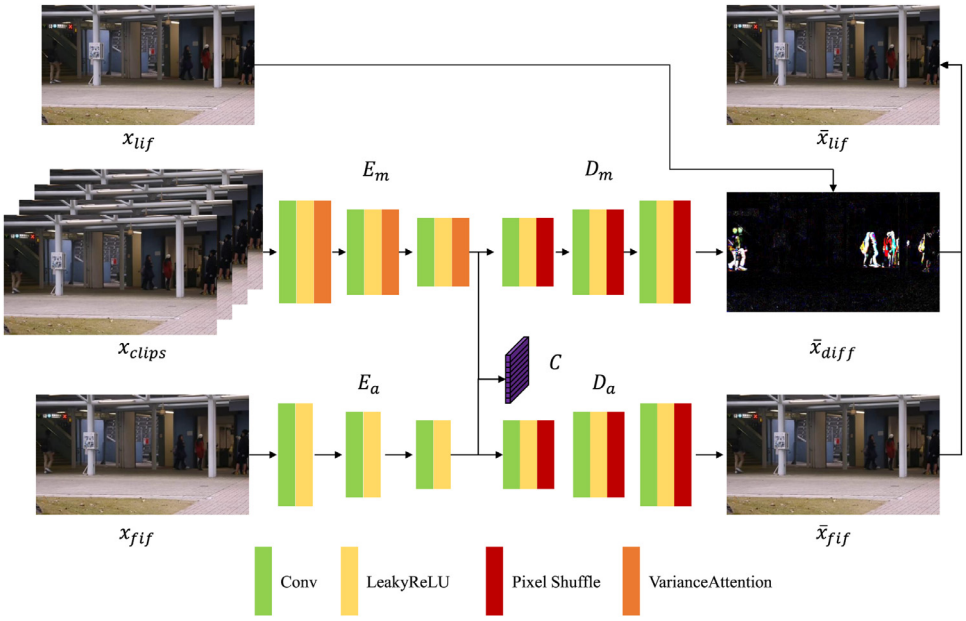


Fig. 1. Overview of our video anomaly detection architecture. We dissociate the spatial-temporal information into two sub-modules. The spatial autoencoder E_a and D_a are used to reconstruct the LIF, while the temporal autoencoder E_m and D_m are applied to predict the RGB difference between the FIF and the LIF with the input consecutive video frames. Both encoders and decoders are constructed by three ResNet blocks. Specifically, we replace the ReLU layers with LeakyReLU in all blocks, and for the decoder networks, we replace the stride convolution layer with pixel shuffle layer to progressively increase the spatial resolution. To further constrain the two streams, we introduce a deep K-means cluster strategy to extract compact representations, represented as the purple area. During the training stage, we optimize the two streams with the deep K-means cluster method according to the distance between the concatenated representations from the spatial encoder and the motion encoder with their corresponding cluster centers. Furthermore, we exploit a variance based attention module which can automatically assign an importance weight to the moving part of video clips in the motion autoencoder. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and then compares the prediction with future frames to identify abnormal events.

However, these approaches mainly focus on learning the motion information and the temporal regularity, missing one crucial factor that the appearance abnormality cue, which is also important. This makes them insensitive to some anomalous objects, which are obviously different from the normal objects in appearance while do not involve motion outliers [8].

Since abnormal events can be irregular in either appearance or motion, it is desirable to combine both the spatial and temporal features for anomaly detection [2,9]. In this work, we decouple the spatio-temporal information into two sub-modules to learn regularity in both the spatial and temporal feature spaces.

Given the consecutive video frames, the spatial autoencoder operates on the first individual frame (FIF), and the motion autoencoder conducts on the first four video frames. The spatial autoencoder, in the form of individual frame appearance, carries information about the scene and objects depicted in the video, while the motion autoencoder produces the RGB difference between the last video frames (LIF) and the FIF to obtain motion information. Then we combine the reconstruction result from the spatial autoencoder with the RGB difference from the temporal autoencoder to get the final prediction. As shown in Fig. 1, our two sub-modules can simultaneously learn the appearance and motion regularity. No matter the event is irregular in the appearance feature space or the motion feature space, it will achieve a large reconstruction error.

In particular, previous works [7,10,11] also exploited a two-stream architecture for anomaly detection, their motion stream learns the motion representation mainly by generating or reconstructing the corresponding optical flow. However, optical flow may not be optimal for learning regularity as they are not specifically designed for anomaly detection [1,12]. Moreover, optical flow estimation has a high computational cost. To overcome these drawbacks, our motion autoencoder takes consecutive video frames as the input and their RGB difference as the output to learn motion information [13], where the RGB difference cue can be obtained much faster than the optical flow to capture the motion information, and the generation of motion autoencoder can be easily pixel-wise fused with the reconstruction of the spatial autoencoder to further help anomaly detection.

Noticeably, most part of the surveillance video is still and outliers usually have a high correlation to fast movement, such as the pedestrian running quickly at the subway entrance and the vehicle driving fast on the pedestrian walkways. Therefore, we exploit a variance based attention module to automatically highlight the image area of large movement and attach this attention module after each block of the motion encoder.

In addition, similar to the previous work [14] which clusters the normal training samples into k clusters by using the K-means algorithm [15], we introduce a deep K-means cluster strategy to force the spatial encoder and the temporal encoder to obtain a more compressed data representation. Specifically, we initialize our cluster centers with the K-means algorithm, and simultaneously optimize the cluster centers and the two streams. By minimizing the distance between the data representation and the cluster centers, normal examples are closely mapped to the cluster centers while anomalous examples are mapped away from the cluster centers.

In brief, our approach considers both the appearance and motion features based on the perception that compared with the normal behavior, an abnormal behavior differs either or both in their appearance and motion patterns.

In summary, our work makes the following contributions:

- We propose a novel autoencoder architecture to dissociate the spatio-temporal representation and learn the regularity in both the spatial feature space and the motion feature space to detect abnormal event in videos.
- We design an efficient motion autoencoder, which takes consecutive video frames as input and RGB difference as output

to imitate the movement of optical flow. The proposed method is much faster than the optical flow-based motion representation learning approach, where its average running time is 32FPS with one GPU.
- We exploit a variance attention module to automatically assign an importance weight to the moving part of video clips, which is useful to improve the performance of the motion autoencoder.
- We explore a deep K-means cluster strategy to force the autoencoder network to generate compact motion and appearance descriptors. Since the cluster is only trained on normal events, the distance between the cluster and the abnormal representation is much higher than that between the normal pattern. The reconstruction error and the cluster distance are together used to evaluate the anomaly.

This paper is an extension of our conference work [16], where the new contributions include:

- We replace the multiple RGB difference output of the original motion autoencoder with the residual between the first and the last individual frame, to make the motion autoencoder learn the longest-range temporal information within the input video frames. Experimental results show that by learning to predict this motion cue is able to improve the performance of the anomaly detection.
- To learn the normality in both the spatial and motion feature spaces, we concatenate these representations extracted from the two streams at the same spatial location, and optimize the two streams and the deep K-means cluster jointly with the early-fusion strategy. Besides, we conduct more experiments to demonstrate the effectiveness of the proposed deep K-means cluster method.
- We modify the anomaly score calculation scheme to fuse the spatio-temporal information with their distance from the deep K-means cluster in the pixel-level. Compared with our prior frame-level fusion scheme, experimental results show that the performance of the new architecture is improved.

The rest of this paper is organized as follows: we first discuss the related work about anomaly detection in Section 2 and then present our proposed architecture in Section 3. Experiments are conducted and analyzed in Section 4. Section 5 concludes this work.

2. Related work

2.1. Anomaly detection with autoencoder

Due to the complexity of realistic data and the limitation of effective labeled data, the anomaly event detection task is usually formulated in an unsupervised setting, where the training sets contain only the normal events.

Most deep learning-based methods use autoencoder [17–20] to extract feature representation, and adopt reconstruction based or prediction based approaches to learn the normality behind the video sequences. The reconstruction based anomaly detection approach takes the given video frames as input and learn to reconstruct normal event with small reconstruction error by extracting the high-level feature representation[1]. applies 2D convolutional autoencoder to reduce dimensionality and learn temporal regularity[21–23]. use the temporal coherency prior of adjacent frames to train an autoencoder network[24]. introduces label-free supervision, which uses constraint learning combined with physics and domain knowledge, to jointly solve three computer vision tasks, including tracking objects and walking man[5]. uses the encoder LSTM to extract features and applies the decoder LSTMs for reconstruction, where this strategy has been widely used for sequential data modeling.

Except the reconstruction based approach, future frame prediction [3] is an alternative deep learning-based method which regards anomaly as the event that does not conform to the expectation. These methods are trained to predict the future frame on the normal training dataset based on its historical observation, and in the testing phase, the abnormal events can be identified by comparing the prediction with their expectation.

We also apply the autoencoder as a backbone network and train it on the normal dataset to extract the common factor. Significantly, we incorporate both the reconstruction-based and the prediction-based architectures, and simultaneously reconstruct the input single frame to capture the appearance feature, and predict the RGB difference between the future frame and the first input frame to learn the motion pattern of the normal event. Consequently, anomalous samples which contain irregular factors in the feature space cannot be reconstructed accurately.

2.2. Video tasks with two stream networks

To fully use both the spatial and temporal information for video tasks, [25] firstly exploits a two-stream network i.e. a RGB-stream and an optical flow-stream, in which the two streams are combined by late fusion for action classification[26]. proposes a spatial-temporal attention module with two network branches for activity recognition[8]. jointly models the appearance and dynamics of crowd patterns, and has demonstrated the effectiveness of the two-stream architecture in modeling complex dynamic scenes.

Since abnormal events can be detected by either appearance or motion, [7] introduces the two-stream architecture for anomaly detection in videos. Besides, image patches and dynamic motion represented by optical flow are employed as input for two separate networks to respectively capture the appearance representation and the motion representation, then the anomaly scores of these two streams are combined by late fusion for final evaluation[11]. utilizes two generator networks to learn the normal patterns of the crowd behavior, where a generator network takes the input frames to produce optical flow images, and the other generator network reconstructs frames from the optical flow[10]. uses two processing streams, where the first autoencoder learns the common spatial appearance structure in normal event and the second stream learns its corresponding motion feature represented by optical flow.

Except the methods that directly take video frames and optical flow as inputs, MPEDRNN [27] extracts 2D human skeleton trajectories and feeds these trajectories into encoders, then simultaneously reconstructing the input and predicting the unseen future with two interacting branches, where each branch consists of an encoder-decoder with RNN to detect anomalous human-related events in surveillance videos.

However, for these methods, it is time costly to acquire optical flow [28] or trajectories. In contrast, we exploit an RGB-difference strategy to replace optical flow to simulate motion information, which is much more efficient. Specifically, at the training stage, we stack all the other frames except the LIF and use the 2D CNN as the backbone of the temporal autoencoder to process consecutive video frames. By enforcing the motion encoder to learn compact motion representation and produce the RGB difference, the motion autoencoder can effectively learn the temporal regularity and motion consistency.

2.3. Data representation and data clustering

Many anomaly detection methods [29–33] aim to find a “compact description” within the normal events. Recently, some auto-

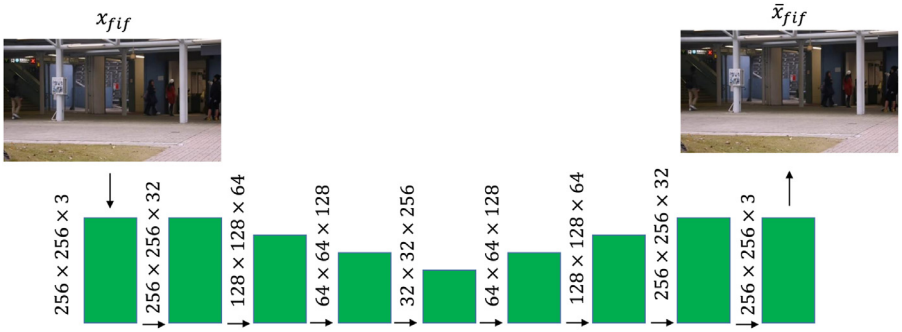


Fig. 2. The structure of our spatial autoencoder with the spatial resolution and the number of channels of feature maps at each bottleneck. We resize all input video frames to 256 × 256 and feed the first frame into the spatial autoencoder. The spatial decoder reconstructs the input frame $\bar{x}_{fif}$ from the spatial representations.

encoder based methods combine feature learning and clustering together[34]. jointly trains a CNN autoencoder and a multinomial logistic regression model to the autoencoder latent space. Similarly, [35] alternates the representation learning and clustering, where a mini-batch k-Means is utilized as the clustering component[36]. proposes a Deep Embedded Clustering (DEC) method, which jointly updates the cluster centers and the data points representations that are initialized from a pre-trained autoencoder. DEC uses soft assignments which are optimized to match stricter assignments through a Kullback-Leibler divergence loss. IDEC [37] and ST-GCAE [38] are subsequently proposed as an improvement of DEC[14]. proposes a supervised classification approach based on clustering the training samples into normality clusters [39]. exploits a memory module for anomaly detection by recording various patterns of normal data into individual items in the memory. Based on this architecture and inspired by the idea of [40], we introduce a deep K-means cluster to force the autoencoder network to generate compact feature representation for video anomaly detection. At the training stage, we train our deep K-means cluster by minimizing the distance between the data representation and the cluster centers. Hence each cluster center can be deemed as a normal spatial-temporal pattern within the training dataset. At the inference stage, the representation of normal samples will be mapped more closely to the cluster centers.

3. Methods

3.1. Overview

For the abnormal event detection task, the training sets contain only the normal events, therefore an effective solution is to learn the regularity in normal training videos with an unsupervised setting. In our proposed method, we dissociate the spatial information and the motion information with a two-stream architecture, and utilize both reconstruction and prediction as the auxiliary task respectively for the spatial stream and the motion stream.

As shown in Fig. 1, there are three main components in our framework to process the given video clips x:

(1) The spatial encoder E_a takes the first individual frame x_{fif} as input and generates the spatial representation z_a, which carries information about the scene and objects depicted in the video. Then we feed z_a into the spatial decoder D_a to get the reconstruction result $\bar{x}_{fif}$.
(2) The motion encoder E_m takes the video clips except the last individual frame x_{lif} as input, denoted as x_{clips}, and we insert the proposed variance-based attention module into the E_m to highlight the fast moving area. The motion decoder D_m is trained to generate the RGB difference x_{diff} between x_{fif} and x_{lif} with the input motion representation z_m, where the generation is denoted as $\bar{x}_{diff}$.
(3) The deep K-means cluster minimizes the distance between the concatenated representation r and the cluster centers C to force both the spatial encoder and the motion encoder networks to extract the common factors within the training sets.

To detect whether the given video clips x is abnormal or not, we compare the final prediction result $\bar{x}_{lif}$ (i.e., the summation of the reconstruction $\bar{x}_{fif}$ and the generated RGB difference $\bar{x}_{diff}$) with x_{lif} to measure the prediction quality. Eventually, we fuse the prediction quality with their distance from the cluster to get the final anomaly score.

3.2. Spatial autoencoder

Since some abnormal objects are partially associated with particular objects, the static appearance of itself is a useful clue [25]. To detect the abnormal object with spatial features such as scene and appearance, we feed the first frame of the input video clips into the spatial autoencoder network. In our model, the spatial encoder is used to encode the input frame to a mid-level appearance representation, and the spatial autoencoder is trained by minimizing the reconstruction error between the input frame x_{fif} and the output frame $\bar{x}_{fif}$, therefore, the bottleneck latent-space z_a contains essential spatial information for frame reconstruction.

Given an individual frame, the spatial encoder converts it to appearance representation z_a, and the spatial decoder generates the reconstruction result $\bar{x}_{fif}$ from the appearance representation z_a:

$$z_a = E_a(x_{fif}; \theta_e^a) \tag{1}$$

$$\bar{x}_{fif} = D_a(z_a; \theta_d^a) \tag{2}$$

where θ_e^a represents the spatial encoder's parameters, θ_d^a denotes the spatial decoder's parameters. Fig. 2 depicts the main structure of our spatial autoencoder. Both the encoder E_a and the decoder D_a are constructed by three ResNet blocks [41]. For the encoder, we remove the two batchnorm layers [42] within each block and attach the batchnorm layer after the block. While for the decoder network, instead of using the deconvolution layer [43] to progressively increase the spatial resolution, we replace the downsampling layer in the ResNet block with the pixel shuffle layer [44] to reduce checkerboard artifacts [45]. Furthermore, we replace the ReLU layers [46] with LeakyReLU for all blocks.

To train the spatial autoencoder to learn regularity in the appearance feature space, we calculate the mean square error between the input x_{fif} and the reconstruction $\bar{x}_{fif}$, where the loss function of the spatial autoencoder l_a is defined as:

$$l_a = \left\| x_{fif} - \bar{x}_{fif} \right\|_2 \tag{3}$$

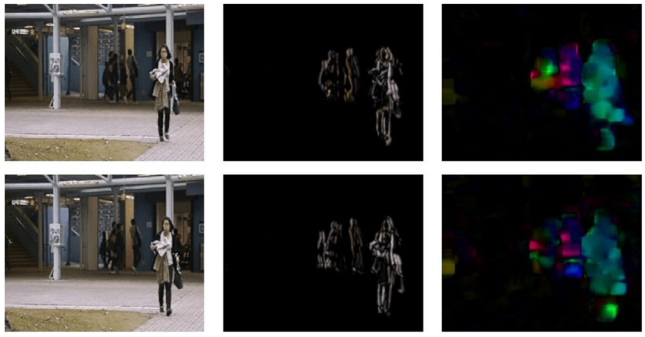


Fig. 3. Some examples of RGB video frames, RGB difference and optical flow.

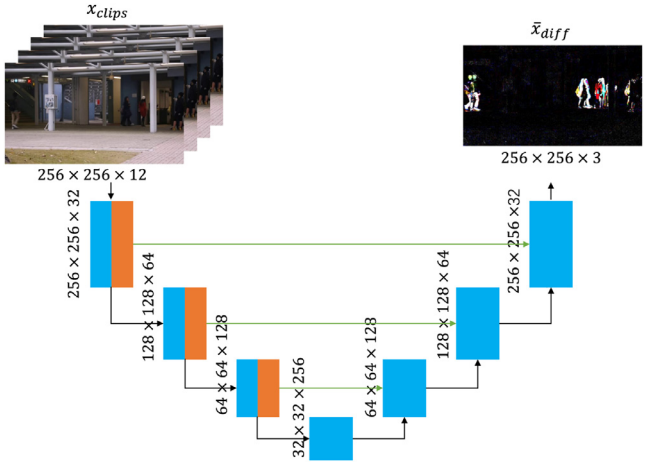


Fig. 4. The structure of our motion autoencoder with the spatial resolution and the number of channels of feature maps at each bottleneck. The input of the motion autoencoder is 4 stacked frames with the size of 256 × 256 × 12. The motion decoder produces the RGB difference $\bar{x}_{diff}$ from the motion representations.

3.3. Motion autoencoder

Most two-stream based convolutional networks utilize warped optical flow as the source for motion modeling [25,47]. Despite the motion feature is very useful, expensive computational cost of optical flow estimation impedes the method, which relies on optical flow, to be used for many real-time implementations.

Inspired by [13], we exploit a novel motion representation to simulate the motion of optical flow, which is directly obtained by the difference of RGB values between video frames. As shown in Fig. 3, it is reasonable to hypothesize that the motion representation captured from optical flow could be learned from the simple cue of RGB difference [13]. Accordingly, we build a motion autoencoder to generate RGB difference with the input of consecutive video frames. By imitating the movement of optical flow with the produced RGB difference, the motion autoencoder can learn the temporal regularity, and its captured feature representation contains essential motion information. For the given video clips x, we stack all the other frames except the LIF as the input, and the RGB difference between the last video frame and the first frame as the target.

Fig. 4 depicts the main structure of our motion autoencoder. We adopt the U-Net [48] architecture and use the 2D CNN as the backbone of the motion autoencoder to process consecutive video frames x_{clips}. The motion encoder E_m converts x_{clips} to motion representations z_m. The motion decoder produces the RGB difference $\bar{x}_{diff}$ from the motion representations:

$$z_m = E_m(x_{clips}; \theta_e^m) \tag{4}$$

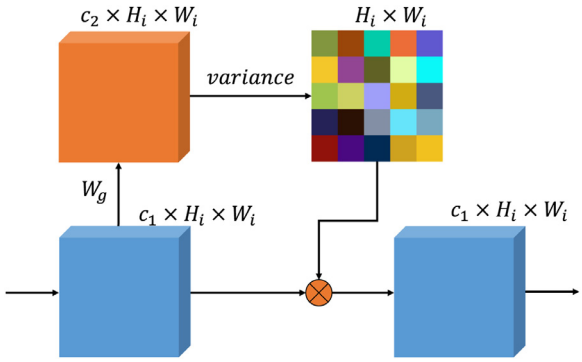


Fig. 5. The structure of the variance attention module. We get the embedding of the input motion features and calculate the variance along the feature dimension followed by operating the l_2 normalization along the spatial dimension to generate the corresponding attention map.

$$\bar{x}_{diff} = D_m(z_m; \theta_d^m) \tag{5}$$

where θ_e^m represents the motion encoder's parameters, θ_d^m denotes the motion decoder's parameters. Similar to the spatial autoencoder, both the encoder E_m and the decoder D_m consist of three ResNet blocks, we remove the batch normalization layer [42] and replace the ReLU layers [46] with LeakyReLU in all blocks. Remarkably, different from the spatial encoder, since there are skip-connections between the motion encoder and the motion decoder, each block in the motion decoder processes both the concatenation of the upsampled motion representation and the low-level feature. To prevent over-smoothing of the generated result, we calculate both the l_2 distance and the gradient loss between the generated RGB difference $\bar{x}_{diff}$ and the ground-truth x_{diff} to make the result more close to the target. The loss function l_m of the motion autoencoder is defined as:

$$l_m = \left\| x_{diff} - \bar{x}_{diff} \right\|_2 + \sum_{d \in \{x,y\}} \left\| \left| g_d(x_{diff}) \right| - \left| g_d(\bar{x}_{diff}) \right| \right\|_1 \tag{6}$$

where g_d denotes the image gradient of video frames along the spatial x-axis and the y-axis. In other words, the final loss function of the motion autoencoder l_m is defined as the combination of the l_2 loss and the gradient loss.

3.4. Variance attention module

It is a normal phenomenon that most part of the surveillance video is static, and the abnormal behaviors are more likely to have large movement changes. Based on this characteristic, we design a variance-based attention in temporal autoencoder to automatically assign the importance weight to the moving part of video clips.

Since our motion encoder consists of three 2D ResNet blocks, every location of the feature maps contains the local motion information across the channels. It is similar to 3D convolution which contains the motion information along the temporal axis, while the 2D convolution contains these information within the feature channels. Therefore, for those areas with large movement, the variance of these embedding will also be higher. Accordingly, we can directly calculate the mean of the representations across the channels and then compute the variance at every location.

Given the embedded motion feature z_m^i from the $i-th$ block of the motion encoder, the attention module firstly feeds the embedded feature into a convolutional layer:

$$f_n(h, w) = W_g * z_m^i(h, w) \tag{7}$$

where $h \in (0, H_i]$ and $w \in (0, W_i]$. H_i and W_i denote the number of rows and columns of the feature maps of the $i-th$ block of the

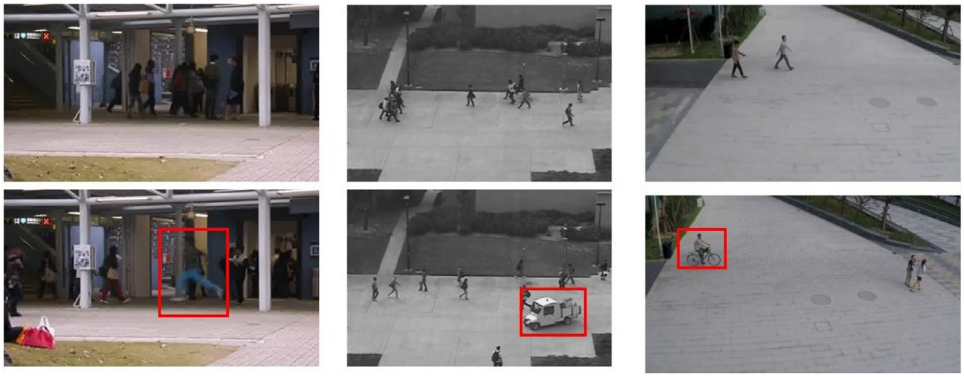


Fig. 6. The first row shows some normal samples and the second row shows some anomaly samples from the CUHK Avenue (left column), the UCSD (middle column), and the ShanghaiTech datasets (right column) respectively. Red boxes denote anomalies in abnormal frames.

motion encoder respectively. W_g represents the weight parameters of the convolutional filter, and we use this convolutional filter to get the embedding of the input features. We calculate the variance along the feature dimension followed by operating the l_2 normalization along the spatial dimension to generate the corresponding attention map g_n:

$$v(h,w) = \frac{1}{D}\sum_{d=1}^{D}\left\| f_n(h,w,d) - \frac{1}{D}\sum_{d=1}^{D} f_n(h,w,d)\right\|_2 \tag{8}$$

$$att(h,w) = \left\| \frac{exp(v(h,w))}{\sum_{h=1,w=1}^{H,W} exp(v(h,w))}\right\|_2 \tag{9}$$

where $v(h,w)$ denotes the variance of the feature maps at the spatial location (h,w). Because the variance-based attention module can highlight the fast moving area, and simultaneously suppress the irrelevant static regions, these abnormal objects which have high correlation with fast moving, i.e. a pedestrian running fast at the subway entrance, will get larger motion loss. This is helpful for the fast moving abnormal event detection. Experiments demonstrated that the proposed variance-based attention is an effective way to amplify the motion loss.

3.5. Clustering

Since we train our motion autoencoder and spatial autoencoder only on the normal data for anomaly detection, the autoencoder may also be generalized on the abnormal events. Therefore, it is essential to push the spatial encoder and the motion encoder to obtain compressed data representation. Inspired by [14,49], we introduce a deep K-means cluster, which minimizes the distance between the data representation and the cluster centers to force both the spatial encoder and the motion encoder networks to extract the common factors within the training sets.

Let K denotes the number of clusters, c_k denotes the representation of cluster k, $1 < k < K$, and $C = \{c_1, ..., c_K\}$ is the set of representations. Given the extracted motion representation z_m and the spatial representation z_a, we firstly concatenate these representations in the feature channels at the same spatial location to fuse the two streams, and train the motion representation and the appearance representation together. Then we normalize the concatenated representation within [0,1], denoted as r. For the representation $r_{h,w} \in R^D$ extracted from the spatial location $h \in (0, H_r], w \in (0, W_r]$, where H_r and W_r denote the spatial size of the concatenated feature maps, we compute the Euclidean distance between the embedding descriptors and each cluster center c_k.

To constitute a continuous generalization of the clustering objective function, we adopt the soft-assignment to calculate the distance between the data representation z_i and the cluster centers C, where the distance is computed as:

$$D(r_{h,w}, C) = \sum_{k=1}^{K} \frac{e^{-\alpha\left\|r_{h,w}-c_k\right\|_2}}{\sum_{k=1}^{K} e^{-\alpha\left\|r_{h,w}-c_k\right\|_2}} \left\|r_{h,w} - c_k\right\|_2^2 \tag{10}$$

where the first part in Eq. (10) represents the soft-assignment of representation $r_{h,w}$ at each cluster center c_k, α is a tunable hyperparameter. The objective function of our deep K-means cluster is defined as:

$$L_{cluster} = \sum_{h=1,w=1}^{H_r,W_r} D(r_{h,w}, C) \tag{11}$$

To initialize the cluster centers, we first train the motion autoencoder and the spatial autoencoder to produce meaningful representations with combination of the spatial loss l_a and the motion loss l_m. After pretraining the two autoencoders, we randomly select several motion representations and their corresponding spatial representations, then we apply the K-means algorithm on the concatenated representations to obtain the initial clustering values.

The deep K-means cluster is also trained on the training set that contains only normal events, each cluster center can be deemed as a certain kind of normality within the training datasets. The anomaly events on the testing set will not affect the cluster centers. During the anomaly event detection, the cluster center will no longer be optimized. As can be seen in Section 4.7, by adding such clustering term to the architecture, the spatial and motion autoencoders can extract more compact representations.

3.6. The training objective function

To learn the model parameters, we combine all the loss functions into our objective function: the spatial loss L_a constrains the model to produce the normal single frame, the motion loss L_m constrains the model to compute the RGB difference between the input video frames and the LIF, and the cluster loss $L_{cluster}$ forces both the motion and spatial autoencoders to minimize the distance between the data representation and the cluster centers:

$$Loss = L_a(x_{fif}, \bar{x}_{fif}) + L_m(x_{diff}, \bar{x}_{diff}) + \lambda_r * L_{cluster} \tag{12}$$

3.7. Anomaly score

The quality of the predicted frame $\bar{x}_{lif}$ generated by $\bar{x}_{fif} + \bar{x}_{diff}$ can be used for anomaly detection, and we compute the Euclidean distance between x_{lif} and $\bar{x}_{lif}$ over all pixel positions to measure the quality of prediction. We also measure the distance between the corresponding concatenated representations r and the cluster centers C because each of them can be deemed as the normality. We calculate the distance over all spatial locations according to

Eq. (10), and then upsample the loss map to the input size using the nearest interpolation, denoted as D_u. For a given test video sequence, we define an anomaly score as the multiplication of their prediction quality and the cluster distance:

$$s = \frac{1}{D_u \left\| x_{lif} - \bar{x}_{lif} \right\|_2} \tag{13}$$

High score indicates that the input video clips are more likely to be normal. Followed by [1], after calculating the score of each video sequence over all spatial locations, we normalize the loss to get a score $S(t)$ in the range of [0,1] for each video frame:

$$S(t) = \frac{s - min_t(s)}{max_t(s) - min_t(s)} \tag{14}$$

We use this normalized score $S(t)$ to evaluate the probability of anomaly events contained in the video clips.

4. Experiments

4.1. Video anomaly detection datasets

We evaluate our model on three publicly available datasets: the UCSD pedestrian dataset [8], the Avenue dataset [50], and the ShanghaiTech dataset [3].

The UCSD Pedestrian dataset includes two subsets Ped1 and Ped2. Since some events are labeled as normality in the training set but are considered as anomalous in the testing set, we only choose the Ped2 subset which contains 16 training videos and 12 testing videos with 12 abnormal events at a resolution of 240 × 360. All of these abnormal cases are about vehicles such as bicycles and cars.

The Avenue dataset contains 16 training videos and 21 testing videos which are captured in front of a subway station at a resolution of 360 × 640. All of these abnormal cases are about throwing objects, loitering and running, and under the challenge of camera position and angle change, where the size of the captured human are also changing.

The ShanghaiTech dataset is the currently largest dataset among abnormal event detection datasets. Compared with other datasets, the ShanghaiTech dataset contains 13 different scenes with various light conditions and camera angles. Its training set contains 330 training videos with over 270,000 training frames, and its testing set contains 107 testing videos with 130 abnormal events and various anomaly types at a resolution of 480 × 856.

4.2. Implementation details

We resize all input video frames to 256 × 256 and use the Adam optimizer [51] to train our network on a single NVIDIA GeForce TitanXp GPU. To initialize the motion and spatial cluster centers, we jointly train the spatial and motion autoencoders in normal dataset without the cluster constraint at first by Eq. 3 and Eq. 6. At this stage, we set the learning rate as 1e-4, and train the spatial and motion autoencoders with 50 epochs for the UCSD Ped2 dataset, and 100 epochs for the Avenue dataset and the ShanghaiTech dataset. Then we freeze the spatial and motion autoencoders, and calculate the cluster centers via K-means to cluster the concatenation motion representation and the spatial representation.

After initialization, the training process of our proposed model performs an alternate optimization. We first freeze the cluster centers and train the autoencoder parameters θ via Eq. 12. Then we freeze the spatial and motion autoencoders and optimize the cluster centers by Eq. 11. We initialize the learning rate to 1e-4 and decrease it to 1e-5 at epoch 100 for the autoencoder part, and set the learning rate as 1e-5 to update the cluster centers. At this stage, we alternately train different parts of our network with 100 epochs for the UCSD Ped2 dataset, and 200 epochs for the Avenue dataset and the ShanghaiTech dataset. The final anomaly detection results are directly calculated based on the reconstruction loss according to Eq. 14.

Table 1
AUC of different methods on the Ped2, Avenue and ShanghaiTech datasets.

Algorithm	UCSD Ped2	Avenue	ShanghaiTech
MPPCA [53]	69.3%	-	-
MPPCA+SFA [8]	61.3%	-	-
MDT [8]	82.9%	-	-
MT-FRCN [54]	92.2%	-	-
Conv2D-AE [1]	85.0%	80.0%	60.9%
Conv3D-AE [52]	91.2%	77.1%	-
ConvLSTM-AE [55]	88.1%	77.0%	-
StackRNN [12]	92.2%	81.7%	68.0%
Abati [56]	95.41%	-	72.5%
MemAE [57]	94.1%	83.3%	71.2%
Liu [3]	95.4%	84.9%	72.8%
Nguyen [10]	96.2%	86.9%	-
Our method	96.7%	87.1%	73.7%

4.3. Evaluation metric

Following the prior works [3,8,12,50], we evaluate our method via the measure of area under the ROC curve (AUC). The ROC curve is obtained by varying the threshold of the anomaly score. A higher AUC value means a more accurate anomaly detection result. To ensure the comparability between different methods, we calculate AUC for the frame-level detection [1,12,52].

4.4. Results

In this section, we compare the proposed method with different hand-crafted feature based methods [8,53,54], and the deep feature based state-of-the-arts which including a 2D convolution autoencoder method (Conv2D-AE) [1], a 3D convolution autoencoder method (Conv3D-AE) [52], a convolution LSTM based autoencoder method (ConvLSTM-AE) [55], a stacked recurrent neural network (StackRNN) [12], and a prediction based method [3]. To be consistent with [3], we set $T = 5$. Specifically, our model takes 4 consecutive frames as the motion autoencoder's input and the first frame as the spatial autoencoder's input. We set the cluster number to be 32 for all datasets.

Table 1 shows the AUC results of our proposed method, the hand-crafted feature based approaches, and the deep feature based methods on all the three benchmark datasets. We can see that our method outperforms all of them. In the upper part, compared to the hand-crafted feature based methods [8,53], the result of our method is at least 4.4% more accurate (96.6% vs 92.2%) on the UCSD Ped2 dataset. In the below part, compared to the deep feature based approaches [1,3,12,52,55,57], our method also performs best on all the three datasets. Particularly, the performance of our algorithm is respectively 1.2%, 1.8%, and 0.9% better than [3] on the UCSD Ped2 dataset, the Avenue dataset, and the ShanghaiTech dataset. In contrast to [10], the performance of us is 0.4% better on the UCSD Ped2 dataset and 0.2% better on the Avenue dataset. Besides, our method only uses the RGB difference as the motion cue, which greatly reduces the computational cost of optical flow estimation. Therefore, our method can be much easier implemented for real-time anomaly detection. The running time comparison will be further discussed in Section 4.8.

Fig. 7 shows some qualitative examples of our method. We can find that for a normal frame, the reconstructed future frame tends

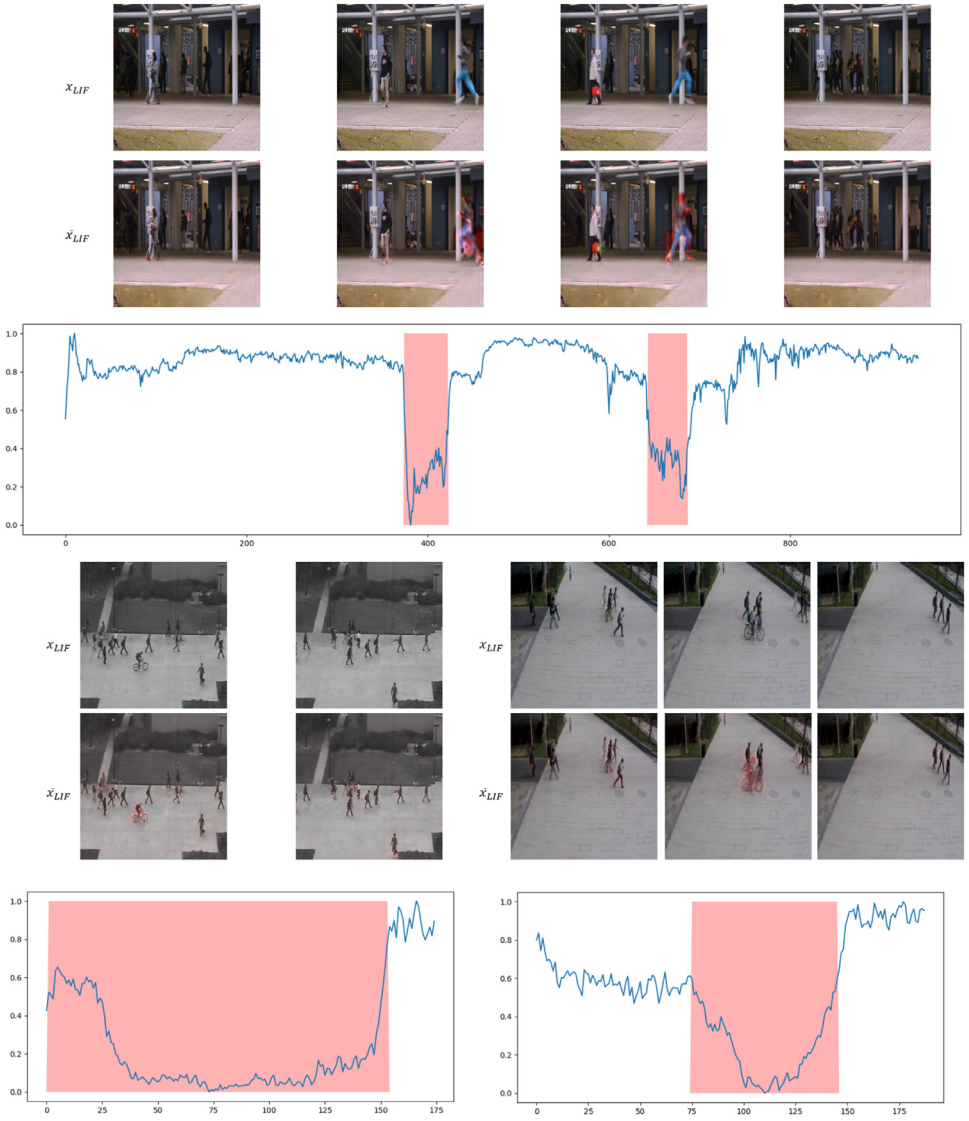


Fig. 7. Part of the temporal regularity score of our method on the Avenue, UCSD Ped2 and ShanghaiTech datasets. The regularity score implies the possibility of normal, and the red shaded regions are the anomaly in groundtruth. Each example shows the groundtruth x_{lif} and the reconstructed $\tilde{(x)}_{lif}$, where the reconstructed $\tilde{(x)}_{lif}$ is superimposed by the reconstruction error map on red channel to denote abnormal areas. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to be close to the actual future prediction. While for an abnormal event, such as a man running across the road on the Avenue dataset, or a bicycle riding on the UCSD Ped2 dataset and the ShanghaiTech dataset, the reconstruction result tends to be blurry or distorted compared with the actual future frame. We superimpose the reconstruction error map on the red channel to further show the abnormal regions on the reconstruction frames.

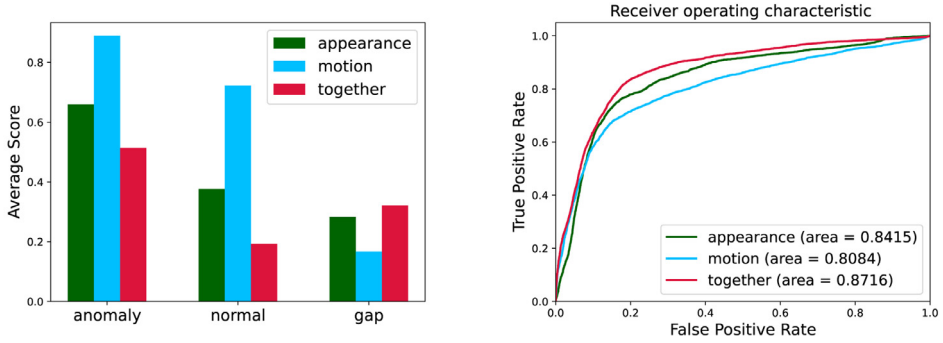
Table 2
Evaluation of different components of our model on the UCSD Ped2 dataset. Results show that the combination of all components gives the best performance.

appearance	√	-	√	√	√	√
motion	-	√	√	√	√	√
variance attention	-	-	-	√	-	√
deep K-means	-	-	-	-	√	√
AUC	91.1%	94.2%	94.7%	95.1%	95.9%	96.7%

4.5. Ablation study

In this subsection, we will conduct several ablation studies to focus on investigating the effect of each component described in Section 3, including the variance attention mechanism, the deep K-means cluster strategy, and the method of combing spatial information and temporal information. We incorporate different components to conduct experiments on the UCSD Ped2 dataset. For the first two studies, we consider only the motion loss and the spatial reconstruction loss. The anomaly score calculation is similar to Eq. 14. For the third study, we consider the reconstruction loss with the variance attention module. For the last study, we consider the full proposed model. Table. 2 validates the effectiveness of each component. We can see that compared with the appearance information, the temporal regularity is more important for video anomaly detection on the UCSD Ped2 dataset. When combining the RGB difference (i.e. motion) with the spatial reconstruction, the performance improves by 0.5% (94.7% vs 94.2%). When the deep K-means cluster constraint is introduced, the performance of the spatio-temporal reconstruction can be further enhanced by 0.7%.

To further illustrate the effectiveness of the appearance and motion information, we calculate the average score of the normal



(a) Average score of normal/abnormal events (b) ROC curves of different cues

Fig. 8. The performance (separability and accuracy) of the appearance and motion cues on the Avenue dataset.

and abnormal events on the Avenue testing set (see Fig. 8a). The corresponding gap, which represents the separability of the normal and abnormal events, is calculated by subtracting the average abnormal score from the average normal score. The larger the gap is, the better the separability. Clearly, the gap of integrating the appearance and motion cues is largest, which demonstrates both the temporal regularity and the appearance cue are useful for detecting abnormal events. Fig. 8b reports the ROC curves of appearance and motion cues on the Avenue dataset. As we can see that different from the performance on the UCSD Ped2 dataset, the appearance information is superior to the motion counterpart where the appearance autoencoder outperforms the motion autoencoder by 3.3%. It means that both the temporal regularity and the appearance cue are important for the video anomaly detection task, and our method can take the advantage of both the motion and spatial information to improve the detection performance.

4.6. Attention visualization

For a deeper understanding on the effect of our variance attention module, we visualize the motion encoder layer of the attention map. For comparison, we also show the input frames. Fig. 9 displays two examples from the Avenue dataset. The left part of Fig. 9 is the normal example, where people walking normally. In the normal scene, the changing part of the video sequence is relatively small, hence the attention weight at each location is consistent. On the other hand, the abnormal event shown in the right part contains a person running across the road. Since the pedestrian moves comparably faster than the other regions over the video, the variance attention module produces higher attention weight to the pedestrian. The corresponding attention map shows that the value in the fast moving area is much higher than the values in other areas. Since the variance attention module can automatically assign the importance weight to the moving part of video clips, the anomaly events (e.g. running) are more likely to cause higher motion loss. Accordingly, the reconstruction of the abnormal object will get larger loss which is helpful for abnormal event detection. Experiments conducted in Section 4.5 demonstrate the effectiveness of the variance attention module.

4.7. Exploration of cluster numbers

To evaluate the performance of the deep K-means cluster strategy on detecting abnormal events in videos, we conduct experiments on removing the deep K-means cluster and changing the number of cluster centers. We use the UCSD-Ped2 dataset for testing and show the AUC results in Table 3. We separately set the number of the cluster centers to be 4, 8, 16, 32 and 64. Since the AUC value obtained by the autoencoder is already high at 95.1%, the cluster constraint can still boost the performance by 1.4% when the number of the cluster centers is set to 32. The AUC results of different size of the cluster centers demonstrate the robustness of our method.

Table 3
AUC of the proposed method with different cluster numbers on the UCSD Ped2 dataset.

cluster numbers	-	4	8	16	32	64
AUC	95.1%	95.2	95.5%	96.0%	96.7%	96.4%

As shown in Fig. 10, we use the t-SNE [58] algorithm to visualize the distribution of the concatenated data representation trained without/with the deep K-means cluster strategy on the UCSD-Ped2 dataset. For these representations, we use their nearest cluster center as the pseudo-label by calculating the Euclidean distance between the representation and each cluster center. Since we optimize both the cluster centers with the two autoencoders during the training stage, compared with the distribution that is trained without the cluster, the representations, which belong to the same cluster center of our model, will be mapped closer to the cluster centers.

4.8. Running time

We compare the accuracy and the efficiency of our RGB difference strategy with optical flow on the UCSD Ped2 dataset. One traditional optical flow algorithm TV-L1 [59] and one deep learning based optical flow method FlowNet2-SD [60] are selected for comparison.

As shown in Fig. 11, our method is about 2.3 times faster than FlowNet2-SD [60]. Specifically, for one video frame, the FlowNet2-SD algorithm costs 0.071 seconds while our RGB difference strategy only needs 0.031 seconds. Furthermore, the accuracy of “RGB+RGB difference” is respectively 2.1% and 2.6% more accurate than “RGB+FlowNet2-SD” and “RGB+TV-L1”.

We implement our method with an NVIDIA GeForce Titan Xp graphics card. It takes 0.0312 seconds to detect the abnormal event per video frame, i.e. 32FPS, which is on par or faster than the state-of-the-art deep learning based methods. For example, the FPS of

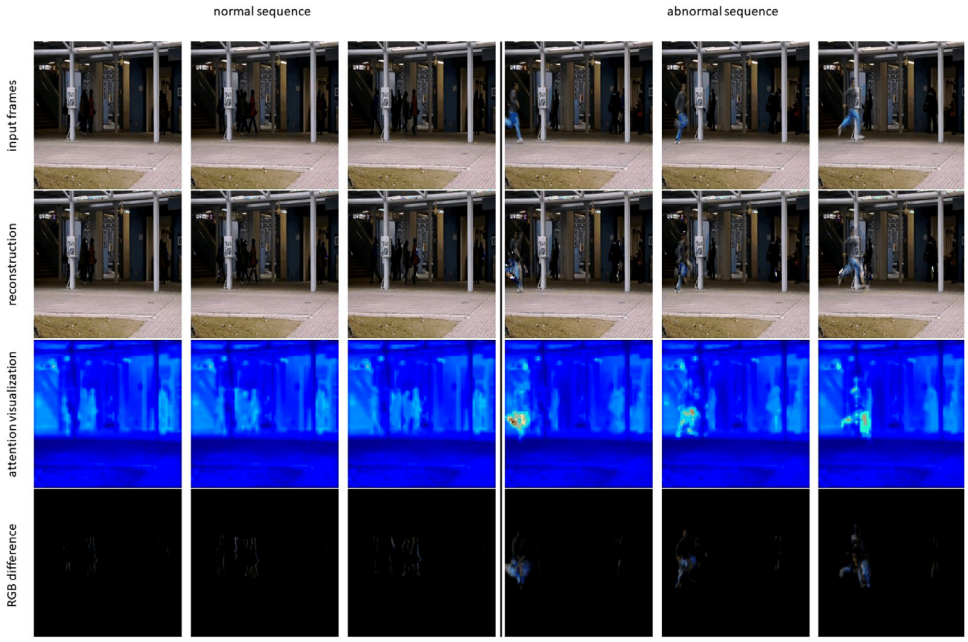


Fig. 9. The first row shows the input video frames, the second row shows the reconstructed frames, and the third row shows the visualization of the attention map in the jet color map. The higher attention weight area is represented closer to red while the lower area is represented closer to blue. The fourth row shows the RGB difference generated from the motion autoencoder. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

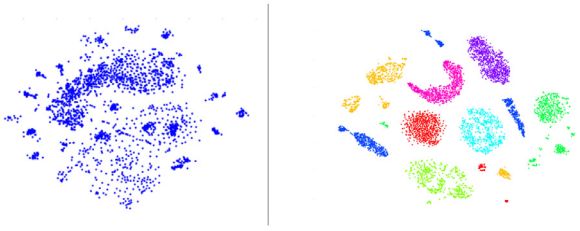


Fig. 10. We use t-SNE [58] visualization for the concatenated data representation to demonstrate the effectiveness of our deep K-means cluster strategy. For the left part, we simultaneously train two autoencoders and randomly selected 5K concatenated representations. For the right part, we train autoencoders with our deep K-means clusters and randomly selected 5K concatenated representations for the 8 cluster centers.

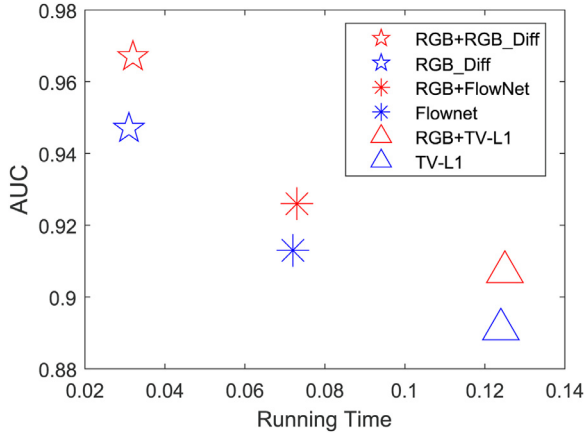


Fig. 11. Result of AUC performance (accuracy) and running time (efficiency) on the UCSD Ped2 dataset. Compared with our "RGB+RGB difference" method to the "RGB+FlowNet" method, the computational time of us is more than 2 times faster, and the AUC performance is improved by 2.1%.

[3], [12], and [61] are respectively 25FPS, 50FPS, and 2FPS. Where the results are copied from these papers originally. Specifically, [12] and [61] respectively are conducted on the extracted feature and the spatio-temporal interest points, while both [3] and our method take the original frame sequence as input.

5. Conclusion

In this paper, we propose a novel autoencoder architecture to dissociate the spatio-temporal information into two sub-modules to learn regularity in both the spatial and temporal feature spaces and to generate the compact description within normal events. Specifically, the spatial autoencoder operates on the first individual frame (FIF) and extracts the regularity in the spatial space by reconstructing the input. The temporal autoencoder processes on the consecutive video frames to learn the temporal regularity by constructing the RGB difference. Depending on the captured temporal regularity and motion consistency, the temporal autoencoder can learn to predict the RGB residual that contains useful motion information for anomaly detection extremely efficient. Furthermore, we design a variance attention module to highlight the moving part of the frame. In addition, to effectively learn the normality in the spatial and motion feature spaces and obtain a more compact data representation, we minimize the distance between the concatenated representation and the cluster centers via a deep K-means cluster method. We combine the result of the spatial autoencoder and the motion autoencoder to obtain the prediction of the last individual frame (LIF), and fuse the prediction with the cluster distance in the pixel-level to evaluate the anomaly. Extensive experiments on three representative datasets demonstrate that our method achieves the state-of-the-art performance.

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.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62106177. It was also supported by

the Wuhan University-Infinova project No.2019010019. The numerical calculation was supported by the supercomputing system in the Super-computing Center of Wuhan University.

References

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

[2] W. Luo, W. Liu, D. Lian, J. Tang, L. Duan, X. Peng, S. Gao, Video anomaly detection with sparse coding inspired deep neural networks, IEEE Trans Pattern Anal Mach Intell (2019), doi:10.1109/TPAMI.2019.2944377. 1–1

[3] W. Liu, W. Luo, D. Lian, S. 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] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, S. Avidan, Graph embedded pose clustering for anomaly detection, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10536–10544, doi:10.1109/CVPR42600.2020.01055.

[5] N. Srivastava, E. Mansimov, R. Salakhudinov, Unsupervised learning of video representations using lstms, in: International conference on machine learning, 2015, pp. 843–852.

[6] F. Tung, J.S. Zelek, D.A. Clausi, Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance, Image Vis Comput 29 (4) (2011) 230–240.

[7] D. Xu, Y. Yan, E. Ricci, N. Sebe, Detecting anomalous events in videos by learning deep representations of appearance and motion, Comput. Vision Image Understanding 156 (2017) 117–127.

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

[9] Y. Zhang, H. Lu, L. Zhang, X. Ruan, Combining motion and appearance cues for anomaly detection, Pattern Recognit 51 (2016) 443–452, doi:10.1016/j.patcog.2015.09.005.

[10] T.-N. Nguyen, J. Meunier, Anomaly detection in video sequence with appearance-motion correspondence, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1273–1283.

[11] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, N. Sebe, Abnormal event detection in videos using generative adversarial nets, in: 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 1577–1581.

[12] W. Luo, W. Liu, S. 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.

[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Trans Pattern Anal Mach Intell (2018). 1–1

[14] 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: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7842–7851.

[15] D. Arthur, S. Vassilvitskii, K-means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7–9, 2007, 2007.

[16] Y. Chang, Z. Tu, W. Xie, J. Yuan, Clustering driven deep autoencoder for video anomaly detection, in: A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision–ECCV 2020, Springer International Publishing, Cham, 2020, pp. 329–345.

[17] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[18] C. Poultney, S. Chopra, Y.L. Cun, et al., Efficient learning of sparse representations with an energy-based model, in: Advances in Neural Information Processing Systems, 2007, pp. 1137–1144.

[19] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ACM, 2008, pp. 1096–1103.

[20] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, Omnipress, 2011, pp. 833–840.

[21] J. Kuen, K.M. Lim, C.P. Lee, Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle, Pattern Recognit 48 (10) (2015) 2964–2982, doi:10.1016/j.patcog.2015.02.012. Discriminative Feature Learning from Big Data for Visual Recognition

[22] V. Ramanathan, K. Tang, G. Mori, L. Fei-Fei, Learning temporal embeddings for complex video analysis, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4471–4479.

[23] P. Wu, J. Liu, M. Li, Y. Sun, F. Shen, Fast sparse coding networks for anomaly detection in videos, Pattern Recognit 107 (2020) 107515, doi:10.1016/j.patcog.2020.107515.

[24] R. Stewart, S. Ermon, Label-free supervision of neural networks with physics and domain knowledge., in: AAAI, 1, 2017, pp. 1–7.

[25] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.

[26] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, G. Francesca, Toyota smarthome: Real-world activities of daily living, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2020.

[27] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, S. Venkatesh, Learning regularity in skeleton trajectories for anomaly detection in videos, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 11996–12004.

[28] Z. Tu, W. Xie, D. Zhang, R. Poppe, R.C. Veltkamp, B. Li, J. Yuan, A survey of variational and CNN-based optical flow techniques, Signal Process. Image Commun. 72 (2019) 9–24.

[29] G. Blanchard, G. Lee, C. Scott, Semi-supervised novelty detection, Journal of Machine Learning Research 11 (2010) 2973–3009.

[30] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International Conference on Machine Learning, 2018, pp. 4393–4402.

[31] M. Turkoz, S. Kim, Y. Son, M.K. Jeong, E.A. Elsayed, Generalized support vector data description for anomaly detection, Pattern Recognit 100 (2020) 107119, doi:10.1016/j.patcog.2019.107119.

[32] P. Perera, R. Nallapati, B. Xiang, Ocgan: One-class novelty detection using gans with constrained latent representations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2898–2906.

[33] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning, Pattern Recognit 58 (2016) 121–134, doi:10.1016/j.patcog.2016.03.028.

[34] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5736–5745.

[35] C. Hsu, C. Lin, Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data, IEEE Trans Multimedia 20 (2) (2017) 421–429.

[36] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International Conference on Machine Learning, 2016, pp. 478–487.

[37] X. Guo, L. Gao, X. Liu, J. Yin, Improved deep embedded clustering with local structure preservation., in: IJCAI, 2017, pp. 1753–1759.

[38] A. Markovitz, G. Sharir, I. Friedman, L. Zelnik-Manor, S. Avidan, Graph embedded pose clustering for anomaly detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10539–10547.

[39] H. Park, J. Noh, B. 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.

[40] M.M. Fard, T. Thonet, E. Gaussier, Deep k-means: jointly clustering with k-means and learning representations, arXiv: Learning (2018).

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi:10.1109/CVPR.2016.90.

[42] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, CoRR abs/1502.03167 (2015).

[43] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 2528–2535, doi:10.1109/CVPR.2010.5539957.

[44] W. Shi, J. Caballero, F. Huszár, J. Totz, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[45] A. Odena, V. Dumoulin, C. Olah, Deconvolution and checkerboard artifacts, Distill 1 (10) (2016).

[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, 25, Curran Associates, Inc., 2012, pp. 1097–1105.

[47] Z. Tu, W. Xie, Q. Qin, R. Poppe, R.C. Veltkamp, B. Li, J. Yuan, Multi-stream cnn: learning representations based on human-related regions for action recognition, Pattern Recognit 79 (2018) 32–43.

[48] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[49] L. Ruff, R.A. Vandermeulen, D. Shoaib, A. Binder, M. Emmanuel, M. Kloft, Deep One-Class Classification(2018).

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

[51] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

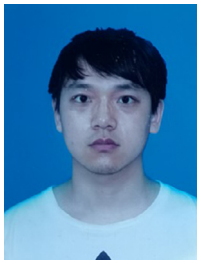
[52] A. Zimek, E. Schubert, H.-P. Kriegel, A survey on unsupervised outlier detection in high-dimensional numerical data, Statistical Analysis and Data Mining: The ASA Data Science Journal 5 (5) (2012) 363–387.

[53] J. Kim, K. 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.

[54] R. Hinami, T. Mei, S. Satoh, Joint detection and recounting of abnormal events by learning deep generic knowledge, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 3619–3627.

[55] W. Luo, W. Liu, S. Gao, Remembering history with convolutional lstm for

anomaly detection, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2017, pp. 439–444.
[56] D. Abati, A. Porrello, S. Calderara, R. Cucchiara, Latent space autoregression for novelty detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 481–490.
[57] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, A.V. Den Hengel, Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection, arXiv: Computer Vision and Pattern Recognition (2019).
[58] L.V.D. Maaten, Accelerating t-SNE using tree-based algorithms, JMLR.org, 2014.
[59] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime tv-l 1 optical flow, in: Joint Pattern Recognition Symposium, Springer, 2007, pp. 214–223.
[60] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, T. Brox, Flownet 2.0: evolution of optical flow estimation with deep networks, IEEE International Conference on Computer Vision (ICCV) (2016), doi:10.1109/CVPR.2017.179.
[61] B. Zhao, L. Fei-Fei, E.P. Xing, Online detection of unusual events in videos via dynamic sparse coding, in: CVPR 2011, IEEE, 2011, pp. 3313–3320.



Yunpeng Chang is currently a Ph.D. student at Wuhan Un iversity. He received the M. Eng. degree at State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing from Wuhan University in 2018. He received the B. Eng. degree from the school o f remote sensing and informa tion enginee ring from Wuhan University in 2015. His research interests mainly include computer vision and machine learning.

Zhigang Tu started his Master Degree in image processing at the School of Electronic Information, Wuhan University, China, 2008. In 2015, he received the Ph.D. degree in Computer Science from Utrecht University, Netherlands. From 2015 to 2016, he was a postdoctoral researcher at Arizona State University, US. Then from 2016 to 2018, he was a research fellow at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently a professor at the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote sensing, Wuhan University. His research interests include computer vision, image processing, video analytics, and machine learning. Special for motion estimation, object segmentation, action recognition and localization, hand/human pose estimation, and anomaly detection. On those topics, he has co-/authored more than 50 articles on international SCI-indexed journals and conferences, e.g. Pattern Recognition (PR), IEEE Trans. Image Process. (TIP), IEEE Trans. Circuits and Sys. Video Tech. (T-CSVT), CVPR, ICCV, ECCV, etc. He is an Associate Editor of The Visual Computer (IF=2.601), a Guest Editor of journals JVCIR (IF=2.678) and Combinatorial Chemistry & High Throughput Screening (IF=1.195). He is the organizer of the ACCV2020 Workshop on MMHAU (Japan) and the ACPR2019 Workshop on MAGR (New Zealand). He serves as a reviewer for more than 10 SCI-indexed journals and conferences, e.g., Pattern Recognition, IEEE Trans. Image Process., IEEE TCSVT, IEEE TMM, CVPR, ICCV, AAAI, ACM MM, etc. He received the “Best Student Paper” Award in the 4th Asian Conference on Artificial Intelligence Technology..

Wei Xie received the B.E. degree in electronic information engineering and the Ph.D. degree in communication and information system fromWuhan University, China, in 2004 and 2010, respectively. From 2010 to 2013, he was an Assistant Professor with the Computer School, Wuhan University. He is currently a Professor with the Computer School, Central China Normal University, China. His research interests include motion estimation, super resolution reconstruction, image fusion, and image enhancement.

Bin Luo received the M.Sc. degree from ENS Cachan, France, in 2003 and the Ph.D. degree from ENST Paris, France, in 2007. From 2008 to 2010, he was a Post-Doctoral Researcher with GIPSALab, France. He is currently a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, China. His research interests include hyperspectral data analysis and high resolution remote sensing image processing.

Junsong Yuan (M’08-SM’14) received his Ph.D. from Northwestern University and M.Eng. from National University of Singapore. He is currently an associate professor at Computer Science and Engineering department of State University of New York at Bu_alo. Before that, he was an associate professor at Nanyang Technological University (NTU), Singapore. His research interests include computer vision, video analytics, gesture and action analysis. He received best paper award from Intl. Conf. on Advanced Robotics (ICAR’17), 2016 Best Paper Award from IEEE Trans. on Multimedia, Doctoral Spotlight Award from IEEE Conf. on Computer Vision and Pattern Recognition (CVPR’09), and outstanding EECS Ph.D. Thesis award from Northwestern University. He is currently Senior Area Editor of Journal of Vis. Communications and Image Repres. (JVCI), Associate Editor of IEEE Trans. on Image Processing (T-IP) and IEEE Trans. on Circuits and Systems for Video Technology (T-CSVT). He is Program Co-chair of ICME’18, and Area Chair of CVPR’17/19/20/21, ACM MM’18, ICPR’18, ICIP’18’17, ACCV’18’14 etc. He is a Fellow of IEEE and International Association of Pattern Recognition (IAPR).