Clustering Driven Deep Autoencoder for Video Anomaly Detection

Yunpeng Chang[1], Zhigang Tu[1⋆], Wei Xie[2], and Junsong Yuan[3]

[1] Wuhan University, Wuhan 430079, China
[2] Central China Normal University, Wuhan 430079, China
[3] State University of New York at Buffalo, Buffalo, NY14260-2500, USA
{tuzhigang,changyunpeng}@whu.edu.cn, {xw}@mail.ccnu.edu.cn,
{jsyuan}@buffalo.edu

Abstract. Because of the ambiguous definition of anomaly and the complexity of real data, video anomaly detection is one of the most challenging problems in intelligent video surveillance. Since the abnormal events are usually different from normal events in appearance and/or in motion behavior, we address this issue by designing a novel convolution autoencoder architecture to separately capture spatial and temporal informative representation. The spatial part reconstructs the last individual frame (LIF), while the temporal part takes consecutive frames as input and RGB difference as output to simulate the generation of optical flow. The abnormal events which are irregular in appearance or in motion behavior lead to a large reconstruction error. Besides, we design a deep k-means cluster to force the appearance and the motion encoder to extract common factors of variation within the dataset. Experiments on some publicly available datasets demonstrate the effectiveness of our method with the state-of-the-art performance.

Keywords: video anomaly detection; spatio-temporal dissociation; deep k-means cluster

1 Introduction

Video anomaly detection refers to the identification of events which are deviated to the expected behavior. Due to the complexity of realistic data and the limited labelled effective data, a promising solution is to learn the regularity in normal videos with unsupervised setting. Methods based on autoencoder for abnormality detection [3, 8, 31, 34, 38, 39], which focus on modeling only the normal pattern of the videos, have been proposed to address the issue of limited labelled data.

Since abnormal events can be detected by either appearance or motion, [23] uses two processing streams, where the first autoencoder learns common appearance spatial structures in normal events and the second stream learns its corresponding motion represented by an optical flow to learn a correspondence between appearances and their associated motions. However, optical flow may

⋆ Corresponding author: Zhigang Tu

not be optimal for learning regularity as they are not specifically designed for this purpose [8, 21]. Moreover, optical flow estimation has a high computational cost [33]. To overcome this drawback, we build a motion autoencoder with the stacked RGB difference [36] to learn motion information, where the RGB difference cue can be obtained much faster than the motion cue of optical flow.

In this paper, we decouple the spatial-temporal information into two submodules to learn regularity in both spatial and temporal feature spaces. Given the consecutive frames, the spatial autoencoder operates on the last individual frame (LIF) and the temporal autoencoder conducts on the rest of video frames. In our architecture, the temporal part produces the RGB difference between the rest of video frames and the LIF to get motion information. The spatial part, in the form of individual frame appearance, carries information about scenes and objects depicted in the video.

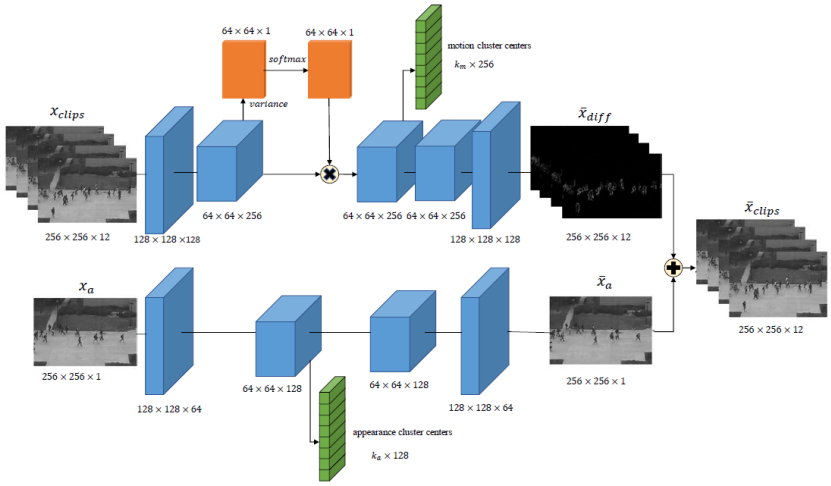

Fig. 1. Overview our video anomaly detection architecture. We dissociate the reconstruction of spatial-temporal information into two independent parts. The spatial part reconstructs the LIF, and the temporal part generates the RGB difference between the rest of video frames and the LIF. Two deep k-means clusters separately force the spatial encoder and the temporal encoder to obtain a more compressed data representation. The orange area represents our variance based attention module which can automatically assign an importance weight to the moving part of video clips in the motion autoencoder.

As shown in Figure 1, our two sub-modules can independently learn appearance and motion features, thus no matter the event is irregular in appearance feature space or motion feature space, the reconstruction of the input consecutive frames will get a large reconstruction error. Based on the characteristic that most part of the surveillance video is still and outliers have a high correlation

to fast moving, we exploit a variance based attention module to automatically assign an importance weight to the moving part of video clips, which is helpful to accelerate the convergence of motion autoencoder.

In addition, we exploit two deep k-means clusters to separately force the spatial encoder and the temporal encoder to obtain a more compressed data representation and extract the common factors of variation within the normal dataset. By minimizing the distance between the data representation and cluster centers, normal examples are closely mapped to the cluster center while anomalous examples are mapped away from the cluster center.

In brief, our approach considers both appearance and motion features based on the perception that compared with normal behaviors, an abnormal behavior differs in their appearance and motion patterns. In summary, this paper makes the following contributions:

- We propose a novel autoencoder architecture to capture informative spatiotemporal representation to detect anomaly in videos by building a novel motion autoencoder, which takes consecutive frames as input and RGB difference as output to simulate the generation of optical flow. Hence the proposed method is much faster than the previous optical flow-based motion representation learning method, where the average running time of our approach is 32fps.
- 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 convergence performance of the motion autoencoder.
- We design a deep k-means cluster 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 representations is much higher than between the normal patterns. The reconstruction error and the cluster distance are together used to assess the anomaly.

2 Related work

2.1 Video Anomaly Detection with Two Stream Networks

Recently, many deep convolutional neural networks [10, 25, 35, 27, 40] have been proposed to extract high-level feature by learning temporal regularity on the video clips. To integrate spatial and temporal information together for video tasks, [30] firstly exploits a two-stream network, i.e. a separate RGB-stream and a optical flow-stream, in which the two streams are combined by late fusion for action classification. [38] introduces the two-stream architecture for anomaly detection. Still image patches and dynamic motion represented by optical flow are employed as input for two separate networks to respectively capture appearance and motion features, and the anomaly scores of these two streams are combined by late fusion for final evaluation. [26] 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 field images, and the other generator network reconstructs frames from the optical flow. However, the time cost of optical flow estimation is expensive [33]. In contrast, we used a RGB-difference strategy to simulate motion information, which is much faster than optical flow.

2.2 Data Representation and Data Clustering

Many anomaly detection methods [2, 18, 28, 29, 24] aim to find a "compact description" within normal events. Recently, several atuto-encoder based methods combine feature learning and clustering together. [5] jointly trains a CNN autoencoder and a multinomial logistic regression model to the autoencoder latent space. Similarly, [11] alternates the representation learning and clustering where a mini-batch k-Means is utilized as the clustering component. [37] proposes a Deep Embedded Clustering (DEC) method, which simultaneously 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 was subsequently proposed in [7] as an improvement of DEC by integrating the autoencoder's reconstruction error in the objective function. [13] proposes a supervised classification approach based on clustering the training samples into normality clusters. Based on this characteristic and inspired by the idea of [4], we design a deep k-means cluster to force the autoencoder network to generate compact feature representations for video anomaly detection.

3 Methods

To address the issues in video based anomaly detection, we introduce a clustering-driven autoencoder to map the normal data into a compact feature representation. Since the abnormal events are different from the normal events in appearance and/or in motion behavior, we decouple our model into two sub-modules, one for spatial part and one for temporal part.

Our proposed autoencoder is composed of three main components: (1) the appearance autoencoder network E_a and D_a, (2) the motion autoencoder network E_m and D_m, and (3) the deep k-means cluster. The spatial part, in the form of individual frame appearance, carries information about scenes and objects depicted in the video. The temporal part, feded the consecutive frames to generate the RGB difference, brings the movement information of the objects. The deep k-means cluster minimizes the distance between the data representation and cluster centers to force both the appearance encoder and the motion encoder networks to extract common factors within the training sets. The main structure of our network is shown in Figure 1.

3.1 Spatial Autoencoder

Since some abnormal objects are partially associated with particular objects, the static appearance by itself is a useful clue [30]. To detect abnormal object with

spatial features such as scenes and appearance, we feed the last frame of input video clips into the sptial autoencoder network. In our model, the appearance encoder is used to encode the input to a mid-level appearance representation from the original image pixels. The appearance autoencoder is trained with the goal of minimizing the reconstruction error between the input frame x_a and the output frame $\bar{x}_a$, therefore, the bottleneck latent-space z_a contains essential spatial information for frame reconstruction.

Given an individual frame, the appearance encoder converts it to appearance representation, denoted as z_a, and the appearance decoder reconstructs the input frame from the appearance representation, denoted as $\bar{x}_a$:

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

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

where θ_e^a represents the set of the encoder's parameters, θ_d^a denotes the set of the decoder's parameters.

The loss function l_a for the appearance autoencoder is defined as Eq.(3):

$$l_a = \|x_a - \bar{x}_a\|_2 \tag{3}$$

3.2 Motion Autoencoder

Most two-stream based convolutional networks utilize warped optical flow as the source for motion modeling [30] [32]. Despite the motion feature is very useful, expensive computational cost of optical flow estimation impedes many real-time implementations. Inspired by [36], we build a motion representation without using optical flow, i.e., the stacked difference of RGB between consecutive frames and the target frame. As shown in Figure 2, it is reasonable to hypothesize that the motion representation captured from optical flow could be learned from the simple cue of RGB difference [36]. Consequently, by learning temporal regularity and motion consistency, the motion autoencoder can learn to predict the RGB residual, and motion autoencoder can extract the data representation that contains essential motion information about the video frames.

We define x_{clips} to denote the consecutive frames, z_m to represent the motion representations, and x_{diff} to represent the RGB difference between the consecutive frames and the LIF, i.e., $x_{diff} = x_{clips} - x_a$. Given the consecutive frames, the motion encoder converts them to motion representations, and each motion representation is denoted as z_m. The motion decoder produces the RGB difference $\bar{x}_{diff}$ from the appearance representations:

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

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

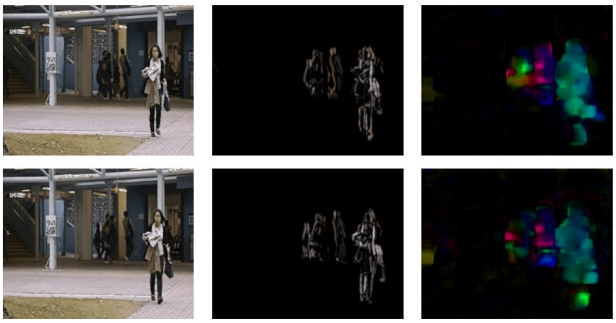

Fig. 2. Some examples of RGB video frames, RGB difference and optical flow.

where θ_e^m represents the set of the encoder's parameters, θ_d^m represents the set of the decoder's parameters. The loss function l_m for the motion autoencoder is given in Eq.(6):

$$l_m = \|x_{diff} - \bar{x}_{diff}\|_2 \tag{6}$$

3.3 Variance attention module

It is obvious that most part of the surveillance video is still, and the abnormal behaviors are more likely to have larger movement changes, thus we aim to learn a function to automatically assign the importance weight to the moving part of video clips. 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. Accordingly, the abnormal object, e.g. pedestrian running fast at the subway entrance, will get larger motion loss which is helpful for fast moving abnormal events detection. Since input video clips contain irrelevant backgrounds, we utilize a temporal attention module to learn the importance of video clips. Given the representation of an input video clip x, the attention module feeds the embedded feature into a convolutional layer:

$$f_n(h, w) = W_g * x(h, w) \tag{7}$$

where $h \in (0, H]$ and $w \in (0, W]$. H and W denote the number of rows and columns of feature maps respectively. W_g represents the weight parameters of convolutional filter. We calculate the variance along the feature dimension followed by operating the l_2 normalization along 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 feature maps at spatial location (h,w).

3.4 Clustering

The role of clustering is to force both the appearance encoder and motion encoder networks to extract the common factors of variation within the dataset. We utilize a deep k-means cluster method to minimize the distance between the data representation and the cluster centers. K is the number of clusters, c_k is the representation of cluster k , $1 < k < K$, and $C = \{c_1, ..., c_K\}$ is the set of representations.

For the motion representation $r_i \in R^D$ extracted from spatial location $i \in \{1, ..., N\}$, we first compute the Euclidean distance between the embeddings descriptors R^D and the corresponding cluster center. To constitute a continuous generalization of the clustering objective function, we adopt the soft-assignment to calculate the distance between the data representation r_i and the cluster centers C, where the distance is computed by Eq.(10):

$$D_m(r_i) = \sum_{k=1}^{K} \frac{e^{-\alpha \|r_i - c_k\|_2}}{\sum_{k=1}^{K} e^{-\alpha \|r_i - c_k\|_2}} \|r_i - c_k\|_2^2 \tag{10}$$

where the first part in Eq.(10) represents the soft-assignment of representation r_i to each cluster center c_k, α is a tunable hyper-parameter.

The cluster center matrix may suffer from redundancy problem if any two cluster centers getting too close. To address this issue, we introduce a penalization term to maxmimize the distance between each cluster. Inspired by [16], we construct a redundancy measure which is defined as dot product of the cluster center matrix C and its transpose C^T, and then subtracting the product by an identity matrix I:

$$R = \left\| CC^T - I \right\|_F \tag{11}$$

where $\|\|_F$ denotes the Frobenius norm of a matrix. This strategy encourages each cluster center to keep the distance from the other cluster centers and punish redundancy within the cluster centers. The objective function of our deep k-means cluster is defined as:

$$L_{cluster} = \sum_{i=1}^{N} D_m(z_i^m, C_m) + \sum_{i=1}^{N} D_a(z_i^a, C_a) + \lambda(R_m + R_a) \tag{12}$$

where D_m and D_a separately represents the distance between motion representations and their cluster centers, and the distance between appearance representations and their cluster centers. R_m and R_a respectively denotes the regularity on the motion cluster center matrix the and appearance cluster center matrix.

Since we optimize the deep k-means cluster on the training sets which contain only normal events, the anomaly events on the test set will not affect the cluster centers. During anomaly event detection, the cluster center will no longer be optimized. Hence the cluster centers can be deemed as a certain kind of normality within the training datasets.

3.5 Training objective

To learn the model parameters, we combine all the loss functions into an objective function to train two autoencoders simultaneously: 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; the cluster loss $L_{cluster}$ forces both motion and spatial autoencoder to minimize the distance between the data representation and the cluster centers:

$$Loss = L_a(x_a, \bar{x}_a) + L_m(x_{diff}, \bar{x}_{diff}) + \lambda_r * L_{cluster} \tag{13}$$

3.6 Anomaly score

We train the model only in normal events, the reconstruction quality of video clips $\bar{x}_{clips}$ generated by $\bar{x}_a + x_{diff}$ can be used for anomaly detection, hence we compute the Euclidean distance between the x_{clips} and the $\bar{x}_{clips}$ of all pixels to measure the quality of reconstruction. The distance between data representation and the closest cluster center is another assessment to qualify the anomaly. For a given test video sequence, we define an anomaly score as:

$$s = \frac{1}{D_m * D_a * \|x_{clips} - \bar{x}_{clips}\|_2^2} \tag{14}$$

High score indicates the input video clips are more likely to be normal. Followed by [8], after calculating the score of each video over all spatial locations, we normalize the losses to get a score S(t) in the range of [0,1] for each frame:

$$S(t) = \frac{s - min_t(s)}{max_t(s) - min_t(s)} \tag{15}$$

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

4 Experiments

4.1 Video anomaly detection datasets

We train our model on three publicly available datasets: the UCSD pedestrian [22], the Avenue [19], and the ShanghaiTech dataset [17]: (1) The UCSD Pedestrian 2 (Ped2) dataset contains 16 training videos and 12 testing videos with 12

abnormal events. All of these abnormal cases are about vehicles such as bicycles and cars. (2) The Avenue dataset contains 16 training videos and 21 testing videos in front of a subway station. All of these abnormal cases are about throwing objects, loitering and running. (3) The ShanghaiTech dataset contains 330 training videos and 107 testing ones with 130 abnormal events. All in all, it consists of 13 scenes and various anomaly types.

Fig. 3. Some samples including normal and abnormal frames in the CUHK Avenue, the UCSD and the ShanghaiTech datasets are used for illustration. Red boxes denote anomalies in abnormal frames.

4.2 Implementation Details

We resize all input video frames to 256 $\times$ 256 and use the Adam optimizer [15] to train our networks. 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 separately cluster the motion representation and 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. 13. Then we freeze the spatial and motion autoencoders and optimize the cluster centers by Eq. 12. For the autoencoder part, we initialize the learning rate to 1e-4 and decrease it to 1e-5 at epoch 100. And we set the learning rate as 1e-5 to update the cluster centers. At this stage, we alternately train different part of our network with 100 epoch 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 both the reconstruction loss and the cluster distance according to Eq. 15.

4.3 Evaluation Metric

Following the prior works [17] [19] [21] [22], we evaluate our method via the area under the ROC curve (AUC). The ROC curve is obtained by varying the threshold of the anomaly score. A higher AUC value represents a more accurate anomaly detection result. To ensure the comparability between different methods, we calculate AUC for the frame-level prediction [43] [8] [21].

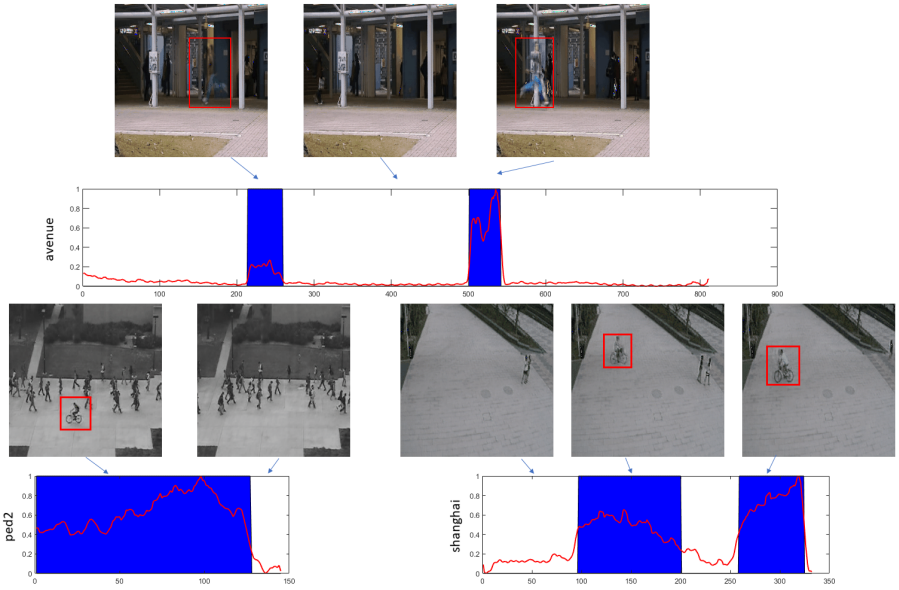

Fig. 4. Parts of the temporal regularity score of our method on the Avenue, UCSD Ped2 and ShanghaiTech datasets. The regularity score implies the possibility of normal, and the blue shaded regions are the anomaly in groundtruth.

4.4 Results

In this section, we compare the proposed method with different hand-crafted feature based methods [14] [22] [9] and deep feature based state-of-the-art methods including a 2D convolution autoencoder method (Conv2D-AE) [8], a 3D convolution autoencoder method (Conv3D-AE) [43], a convolution LSTM based autoencoder method (ConvLSTM-AE) [20], a stacked recurrent neural network (StackRNN) [21], and a prediction based method [17]. To be consistent with [17], we set T = 5. Specifically, our model takes 4 consecutive frames as the motion input and the last frame as the spatial autoencoder's input. We set both the motion cluster number and spatial cluster number to 32 for all datasets.

Table 1 shows the AUC results of our proposed method and the state-of-the-art approaches. We can see that our method outperforms all of them. In the upper part, compared to the hand-crafted feature based methods [14, 22], the result of the proposed method is at least 4.3% more accurate (96.5% vs 92.2%) on the UCSD Ped2 dataset. In the below part, compared to the deep feature based approaches [8, 43, 20, 21, 17, 6], our method also performs best on all the three datasets. Particularly, the performance of our algorithm is respectively 1.1%, 1.1%, and 0.5% better than [17] on the UCSD Ped2 dataset, the Avenue dataset, and the ShanghaiTech dataset. Besides, compared to the latest approach [23], the accuracy of our method is still 0.3% higher on the UCSD Ped2 dataset.

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

Algorithm	UCSD Ped2	Avenue	ShanghaiTech
MPPCA [14]	69.3%	-	-
MPPCA+SFA [22]	61.3%	-	-
MDT [22]	82.9%	-	-
MT-FRCN [9]	92.2%	-	-
Conv2D-AE [8]	85.0%	80.0%	60.9%
Conv3D-AE [43]	91.2%	77.1%	-
ConvLSTM-AE [20]	88.1%	77.0%	-
StackRNN [21]	92.2%	81.7%	68.0%
Abati [1]	95.41%	-%	72.5%
MemAE [6]	94.1%	83.3%	71.2%
Liu [17]	95.4%	84.9%	72.8%
Nguyen and Meunier [23]	96.2%	86.9%	-
Our method	96.5%	86.0%	73.3%

Figure 4 shows some qualitative examples of our method. We can find that for a normal frame, the predicted future frame tends to be close to the actual future prediction. For an abnormal frame, the predicted future frame tends to be blurry or distorted compared with the actual future frame.

4.5 Ablation study

In this subsection, we focus on investigating the effect of each component described in Section 3, including the variance attention mechanism, deep k-means clusters, and the combination of spatial information and temporal information. We combine different part of our components to conduct experiments on the Avenue dataset. For the first two parts, we consider only the motion loss and the spatial reconstruction loss. The anomaly score calculation is similar to Eq. 15. For the third part, we consider the reconstruction loss with the variance attention module. For the last part, 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. When combining the RGB difference with the spatial reconstruction, the performance improves by 2.9%. When the deep k-means cluster constraint is introduced, the spatiotemporal reconstruction multiplied by their cluster distance can further enhance the performance by 3.1%.

Table 2. Evaluation of different components of our architecture on the Avenue dataset. Results show that the combination of all components gives the best performance.

motion	√	-	√	√	√	√
appearance	-	√	√	√	√	√
variance attention	-	-	-	√	-	√
deep k-means	-	-	-	-	√	√
AUC	79.9%	71.2%	81.4%	82.8%	83.5%	86.0%

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

Algorithm	UCSD Ped2
without k-means	94.5%
4	95.6
8	95.5%
16	96.0%
32	96.5%
64	96.4%

4.6 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 deep k-means cluster and changing the number of cluster centers. We use the UCSD-Ped2 datatset for testing and show the AUC results in Tabel 3. We separately set the number of the spatial cluster center and the motion cluster center to be 4, 8, 16, 32. Since the AUC value obtained by the autoencoder is already high at 94.5%, the cluster constraint can boost the performance by 1.1%. The AUC results of different size of cluster centers demonstrate the robustness of our method.

4.7 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. Figure 5 shows two examples from the Avenue dataset. The left part of Figure 5 is the normal example, where people walking

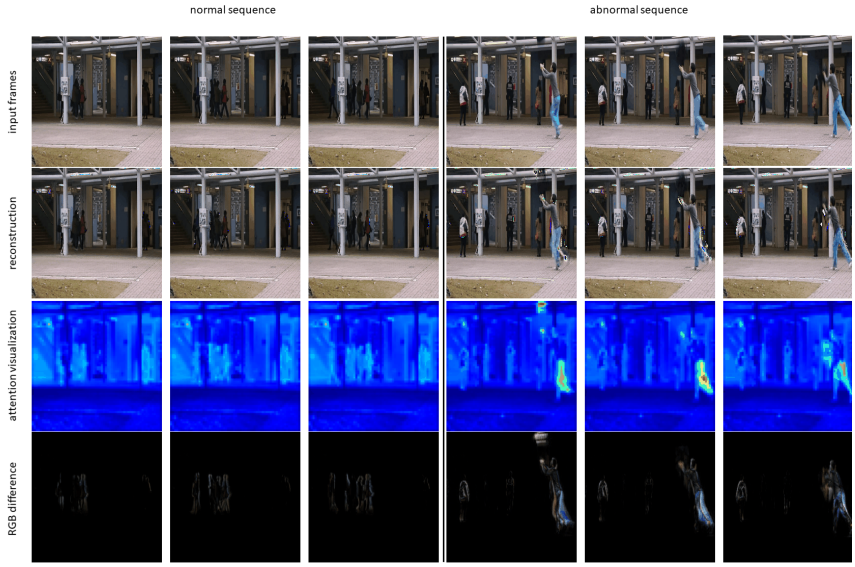

Fig. 5. The first row shows the input video frames, and the second row shows the reconstructed frames. The third row shows the visualization of the attention map in jet color map. The higher attention weight area is represented closer to red while the lower area is represented closer to blue. The forth row shows the RGB difference generated from the motion autoencoder.

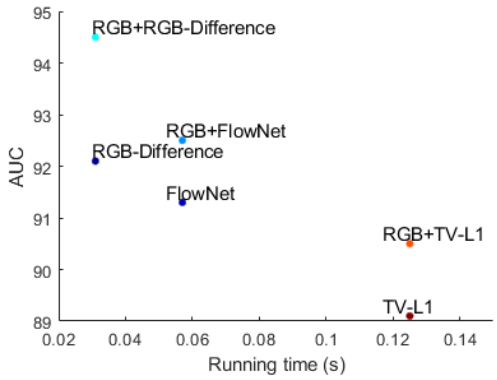

Fig. 6. AUC performance and running time on the UCSD Ped2 dataset. Compared with our "RGB+RGB difference" to the "RGB+FlowNet" method, the computational time of us is about 2 times faster, and the AUC performance is improved by 2.1%.

normally. In the normal scene, the changing part of video sequence is relatively

small, hence the attention weight of each location is quite consistent. On the other hand, the abnormal event contains a person throwing a bag, the variance attention module produces higher attention weight in areas where the movement is fast. The corresponding attention map shows that the value in the thrown bag 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 such as running are more likely to cause higher reconstruction error. The experiments conducted in Section 4.5 demonstrate the effectiveness of the variance attention module.

4.8 Comparison with Optical Flow

We compare the performance and running time of RGB difference with the optical flow on the UCSD Ped2 dataset. One traditional optical flow algorithm TV-L1 [41] and one deep learning based optical flow method FlowNet2-SD [12] are selected for comparison. As shown in Figure 6, our method is about 2.3 times faster than FlowNet2-SD [12]. 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 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 abnormal events per one video frame, i.e. 32fps, which is on par or faster than previous state-of-the-art deep learning based methods. For example, the fps of [17], [21], and [42] are respectively 25fps, 50fps, and 2fps (Where the results are copied from the original corresponding papers).

5 Conclusion

In this paper, we propose a novel clustering-driven deep autoencoder technique to generate the compact description within normal events. To learn regularity in both spatial and temporal feature spaces, we decouple the spatial-temporal information into two sub-modules. Given the consecutive frames, the spatial autoencoder operates on the last individual frame, and the temporal autoencoder processes on the rest of video frames to learn the temporal regularity by constructing the RGB difference. To force both the spatial encoder and the temporal encoder to obtain a more compact data representation, we minimize the distance between the data representation and cluster centers via two deep k-means clusters. Since the cluster is only trained on the normal events, the distance between the cluster and the representations of anomaly events is much higher than between the normal patterns. We use both the reconstruction error and the cluster distance to evaluate the anomaly. Extensive experiments on three datasets demonstrate that our method achieves the state-of-the-art performance.

Acknowledgements. This work was supported by the Fundamental Research Funds for the Central Universities (2042020KF0016 and CCNU20TS028). It was also supported by the Wuhan University-Huawei Company Project.

References

1. Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 481–490 (2019)
2. Blanchard, G., Lee, G., Scott, C.: Semi-supervised novelty detection. Journal of Machine Learning Research **11**, 2973–3009 (2010)
3. Chang, Y., Tu, Z., Luo, B., Qin, Q.: Learning spatiotemporal representation based on 3d autoencoder for anomaly detection. In: Asian Conference on Pattern Recognition. pp. 187–195 (2019)
4. Fard, M.M., Thonet, T., Gaussier, E.: Deep k-means: Jointly clustering with k-means and learning representations. arXiv: Learning (2018)
5. Ghasedi Dizaji, K., Herandi, A., Deng, C., Cai, W., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: IEEE International Conference on Computer Vision (CVPR). pp. 5736–5745 (2017)
6. Gong, D., Liu, L., Le, V., Saha, B., Mansour, M.R., Venkatesh, S., Den Hengel, A.V.: Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: IEEE International Conference on Computer Vision (ICCV). pp. 1705–1714 (2019)
7. Guo, X., Gao, L., Liu, X., Yin, J.: Improved deep embedded clustering with local structure preservation. In: International Joint Conferences on Artificial Intelligence (IJCAI). pp. 1753–1759 (2017)
8. Hasan, M., Choi, J., Neumann, J., Roy-Chowdhury, A.K., Davis, L.S.: Learning temporal regularity in video sequences. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 733–742 (2016)
9. Hinami, R., Mei, T., Satoh, S.: Joint detection and recounting of abnormal events by learning deep generic knowledge. In: IEEE International Conference on Computer Vision (ICCV). pp. 3619–3627 (2017)
10. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science **313**(5786), 504–507 (2006)
11. Hsu, C., Lin, C.: Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. IEEE Transactions on Multimedia **20**(2), 421–429 (2017)
12. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2462–2470 (2017)
13. Ionescu, R.T., Khan, F.S., Georgescu, M.I., Shao, L.: Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 7842–7851 (2019)
14. Kim, J., Grauman, K.: Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2921–2928 (2009)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)
16. Lin, Z., Feng, M., Santos, C.N.d., Yu, M., Xiang, B., Zhou, B., Bengio, Y.: A structured self-attentive sentence embedding. In: International Conference on Learning Representations (ICLR) (2017)
17. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection–a new baseline. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 6536–6545 (2018)

18. Liu, Y., Zheng, Y.F.: Minimum enclosing and maximum excluding machine for pattern description and discrimination. In: International Conference on Pattern Recognition (ICPR). vol. 3, pp. 129–132 (2006)
19. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 fps in matlab. In: IEEE international conference on computer vision. pp. 2720–2727 (2013)
20. Luo, W., Liu, W., Gao, S.: Remembering history with convolutional lstm for anomaly detection. In: International Conference on Multimedia and Expo (ICME). pp. 439–444 (2017)
21. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked rnn framework. In: IEEE International Conference on Computer Vision. pp. 341–349 (2017)
22. Mahadevan, V., Li, W., Bhalodia, V., Vasconcelos, N.: Anomaly detection in crowded scenes. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 1975–1981. IEEE (2010)
23. Nguyen, T.N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: IEEE International Conference on Computer Vision (ICCV). pp. 1273–1283 (2019)
24. Perera, P., Nallapati, R., Xiang, B.: Ocgan: One-class novelty detection using gans with constrained latent representations. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 2898–2906 (2019)
25. Poultney, C., Chopra, S., Cun, Y.L., et al.: Efficient learning of sparse representations with an energy-based model. In: Advances in neural information processing systems. pp. 1137–1144 (2007)
26. Ravanbakhsh, M., Nabi, M., Sangineto, E., Marcenaro, L., Regazzoni, C., Sebe, N.: Abnormal event detection in videos using generative adversarial nets. In: IEEE International Conference on Image Processing (ICIP). pp. 1577–1581 (2017)
27. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: International Conference on Machine Learning (ICML). pp. 833–840 (2011)
28. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International Conference on Machine Learning. pp. 4393–4402 (2018)
29. Ruff, L., Vandermeulen, R.A., Görnitz, N., Binder, A., Müller, E., Müller, K.R., Kloft, M.: Deep semi-supervised anomaly detection. In: International Conference on Learning Representations (ICLR) (2020)
30. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in neural information processing systems. pp. 568–576 (2014)
31. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using lstms. In: International conference on machine learning. pp. 843–852 (2015)
32. Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R.C., Li, B., Yuan, J.: Multi-stream CNN: Learning representations based on human-related regions for action recognition. Pattern Recognition **79**, 32–43 (2018)
33. Tu, Z., Xie, W., Zhang, D., Poppe, R., Veltkamp, R.C., Li, B., Yuan, J.: A survey of variational and cnn-based optical flow techniques. Signal Processing: Image Communication **72**, 9–24 (2019)
34. Tung, F., Zelek, J.S., Clausi, D.A.: Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. Image and Vision Computing **29**(4), 230–240 (2011)

35. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: International conference on Machine learning (ICML). pp. 1096–1103 (2008)
36. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks for action recognition in videos. IEEE Transactions on Pattern Analysis and Machine Intelligence pp. 1–1 (2018)
37. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International conference on machine learning. pp. 478–487 (2016)
38. Xu, D., Yan, Y., Ricci, E., Sebe, N.: Detecting anomalous events in videos by learning deep representations of appearance and motion. Computer Vision and Image Understanding **156**, 117–127 (2017)
39. Yan, M., Meng, J., Zhou, C., Tu, Z., Tan, Y.P., Yuan, J.: Detecting spatiotemporal irregularities in videos via a 3d convolutional autoencoder. Journal of Visual Communication and Image Representation **67**, 102747 (2020)
40. Yu, T., Ren, Z., Li, Y., Yan, E., Xu, N., Yuan, J.: Temporal structure mining for weakly supervised action detection. In: IEEE International Conference on Computer Vision. pp. 5522–5531 (2019)
41. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime tv-l1 optical flow. In: Joint pattern recognition symposium. pp. 214–223 (2007)
42. Zhao, B., Fei-Fei, L., Xing, E.P.: Online detection of unusual events in videos via dynamic sparse coding. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3313–3320 (2011)
43. Zimek, A., Schubert, E., Kriegel, H.P.: A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining **5**(5), 363–387 (2012)