SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Reverse erasure guided spatio-temporal autoencoder with compact feature representation for video anomaly detection

Yuanhong Zhong[1*], Xia Chen[1], Jinyang Jiang[2] & Fan Ren[3]

[1]*School of Microelectronics and Communication Engineering, Chongqing University, Chongqing* 400044*, China;*
[2]*State Grid Chongqing Electric Power Research Institute, Chongqing* 401123*, China;*
[3]*Changan Software Technology Company, Changan Automobile Corp, Chongqing* 401120*, China*

Appendix A Anomaly evaluation

In this paper, I_t represents the t-th frame in a video sequence, and $I_{t-\Delta}$ represents the Δ-th frame before I_t. Given the frames erased from the raw frames $I_{t-\Delta}$ to I_t, the model is trained to reconstruct the t-th frame $\hat{I}_t$ while minimizing the volume of hypersphere that contains the feature representations of reconstruction frames. In the testing phase, given the frames from $I_{t-\Delta}$ to I_t without erasure, the model is to reconstruct the frame $\hat{I}_t$, and the abnormality of I_t is evaluated according to reconstruction error. The calculation process of abnormality score written more compactly in Algorithm A1.

Algorithm A1 Procedure for calculating abnormality score of frame.

Require: I:input frame, $\hat{I}$: reconstruction frame, H: height, W: width, C: channel, k: size of sliding window, s: step of sliding window;
Ensure: $Score$: abnormality score of test frame;
i: the position of pixel in horizontal direction;
j: the position of pixel in vertical direction;
R: set of reconstruction error based on patch;
$E \Leftarrow \sum_{c=1}^{C} \hat{I} - I$: reconstruction error map;
for all sliding windows $[i_{start}, i_{start}+k)$ until W and $[j_{start}, j_{start}+k)$ until H **do**
    $P \Leftarrow E(i,j)$, $i \in [i_{start}, i_{start}+k)$ and $j \in [j_{start}, j_{start}+k)$: the patch in reconstruction error map determined by sliding window;
    add MSE(P) into R, where MSE represents Mean Squared Error;
end for
$Score \Leftarrow \max R$;

Appendix B Datasets

We will briefly introduce the three datasets used in this paper. Some normal and abnormal examples are listed in Figure B1 .

CUHK Avenue dataset. The dataset contains 37 videos, in which 16 videos with 15328 frames are used for training model and the remaining 21 videos with 15324 frames are picked out for evaluating the anomaly detection performance of model. The resolution of each frame is 640 × 360. In this dataset, 47 abnormal events can be observed, including loitering, throwing objects, and running.

UCSD Pedestrian dataset. The dataset contains UCSD Pedestrian 1 dataset and UCSD Pedestrian 2 dataset, denoted as Ped1 and Ped2 respectively. We experiment on Ped2, but not Ped1, because the frame resolution of 158 × 238 in Ped1 is quite low. In Ped2, there are 16 training videos and 12 testing videos, each with no more than 200 frames. The video frame has the resolution of 360 × 240. There are 12 irregular events in Ped2 dataset, mainly objects with abnormal appearance, such as bicycles and trucks on sidewalk.

ShanghaiTech dataset. The dataset is a significantly challenging video anomaly detection dataset, which consists of 13 scenes with over 270000 training frames. It contains 330 training videos and 107 test videos. The resolution of each frame is 856 × 480. There are 130 abnormal events in the ShanghaiTech dataset, including the emergence of bicycles, skateboards and others.

Appendix C Experiments

We evaluate the detection performance of the proposed method as well as the effects of different components by extensive experiments. Experiments are conducted on three publicly available datasets, including CUHK Avenue dataset [1], UCSD pedestrian dataset [2], and ShanghaiTech Campus dataset [3]. The proposed model is implemented using Pytorch [4] and trained with Adam algorithm [5]. The network parameters of the encoder and decoder module included in the proposed model are set as shown in Figure C1. During the training, the cosine annealing [6] decay with initial learning rate of 0.0002 is utilized. The batch size is set

* Corresponding author (email: zhongyh@cqu.edu.cn)

Figure B1 Examples of normal (top) and abnormal (bottom) crowd activities in CUHK Avenue, UCSD Pedestrian 2, and ShanghaiTech datasets, where the abnormal activities are marked by the red bounding boxes.

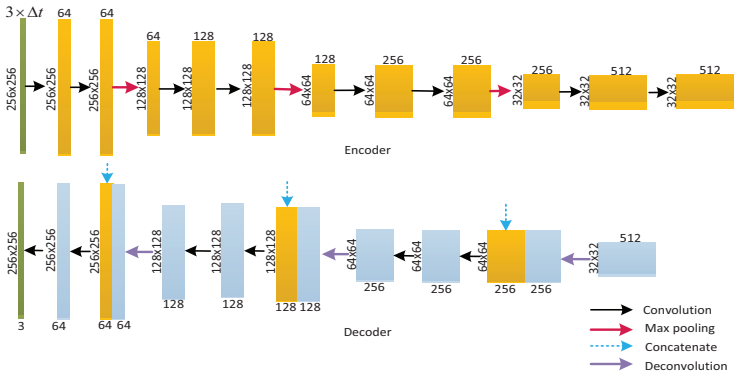


Figure C1 The detailed network architecture of the proposed model, composed of encoder (the upper part) and decoder (the lower part). Note that there are skip connections between the encoder and decoder in prediction stream. Δ_t represents the number of images input to the encoder.

to 4 with the training epochs of 60, 60, and 10 epochs on CUHK Avenue, UCSD Ped2, and ShanghaiTech datasets, respectively. For all datasets, the frame is resized to 256 × 256 pixels and the pixel intensity is normalized to the range of [-1, 1]. The total length of input frames is set to 5, i.e., $\Delta = 4$. It is found empirically that the hyper parameters λ_{int}, λ_{rgb}, and $\lambda_{compact}$ are set as 1, 0.2, 0.01 respectively in the training loss function, which achieves good performance on all datasets. In SVDD, we set v as 0.1 to ensure the tolerance of model to various normal patterns. To reduce the memory required for calculation, this paper does not calculate a special mask for each frame in training set. Instead, we perform an "or" operation on multiple masks latest generated in the last epoch of training to generate a universal mask for erasure in the next training epoch. To evaluate the abnormality of a frame, the size of P was set as 16 × 16, which is determined by sliding a window with step size of 4 in considering frame. Our code is available online:https://github.com/Xia-Chen/DESDnet. The visualization results of model for frame reconstruction on the three datasets are provided in Figure C2. From the reconstruction error map, we can see that normal regions can be reconstructed well, but abnormal regions cannot.

Appendix D Ablation studies

In this section, we analyze different components of the proposed model in detail to explore their advantages, including the fusion of reconstruction and prediction, reverse erasure, and the constraint based on SVDD. These experiments were carried out on Avenue and Ped2 datasets, but not on ShanghaiTech dataset due to its large scale.

Appendix D.1 Effect of the fusion of reconstruction and prediction

To evaluate the effect of fusing reconstruction and prediction, we combine reconstruction encoder and prediction encoder with the decoder to get three different models: 1) the reconstruction model composed of reconstruction encoder and decoder, taking the frame I_t as input; 2) the prediction model composed of prediction encoder and decoder, taking the frames from $I_{t-\Delta}$ to I_{t-1} as input; 3) the proposed model composed of reconstruction encoder, prediction encoder, and decoder, taking the frames from $I_{t-\Delta}$ to I_t as input. In order to be consistent with the proposed model, there are the skip connections between the encoder and decoder in prediction model, but not in reconstruction model. For training each model, the pixel intensity loss, the weighted RGB loss, and the feature compact loss are adopted to supervise the training. Through these models, the performance of reconstruction model and prediction model in detecting anomalies independently can be obtained.

Figure D1 shows the abnormality score of video sequences in Avenue and Ped2 datasets on the above three models. The results show that the proposed model generates large reconstruction errors on abnormal frames and small errors on normal frames. The average of normal and abnormal scores and the gap between them is given in Figure D2. In general, the proposed model achieves the largest score gap on each dataset, indicating the proposed model has better detection performance. Besides that, the AUC

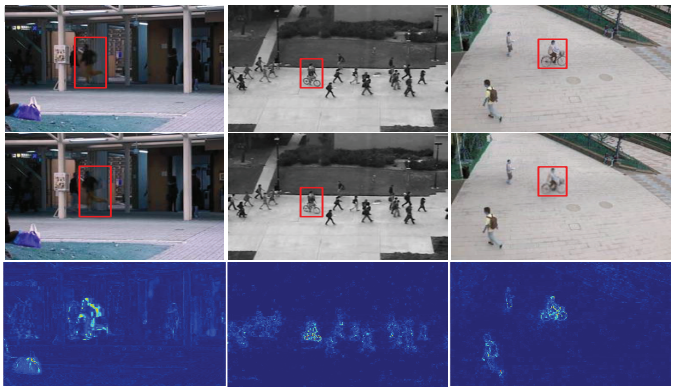
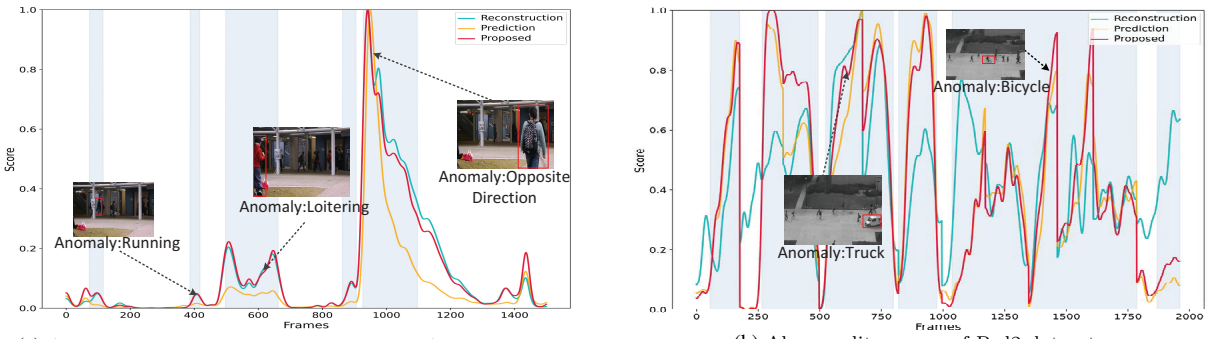


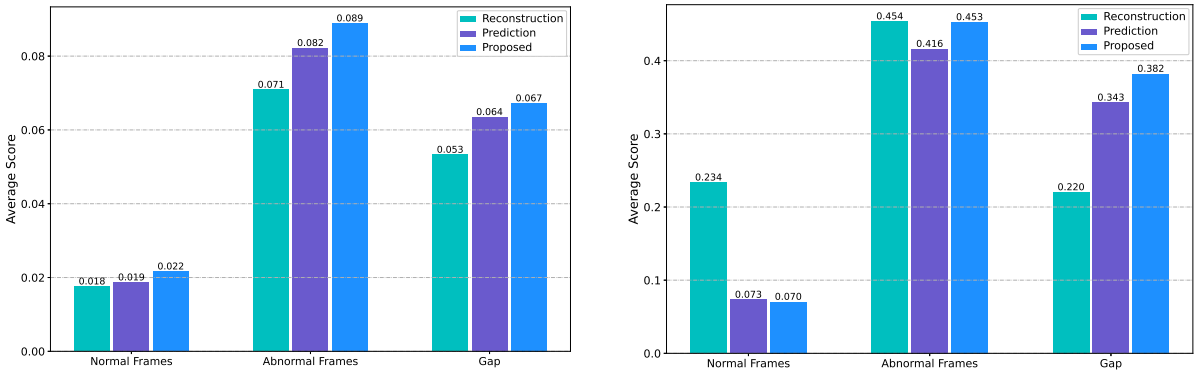
Figure C2 Outputs of the proposed model for several abnormal frames from (left to right) the CUHK Avenue, UCSD Ped2, and ShanghaiTech datasets: input frames (top); output frames (middle); error maps (bottom). Lighter color represents larger error.



(a) Abnormality scores of a sequence in the Avenue dataset

(b) Abnormality scores of Ped2 dataset

Figure D1 Abnormality score comparison on CUHK Avenue and UCSD Ped2 datasets. A high score means abnormal, otherwise normal. Light blue regions represent abnormal frames, and the abnormal events are marked by the red bounding boxes.



(a) Average score of Avenue dataset

(b) Average score of Ped2 dataset

Figure D2 Average score comparison of normal and abnormal frame, where gap represents the average score of abnormal frame minus that of normal frame.

Table D1 AUC/EER comparison of the proposed model with simple reconstruction model and prediction model.

	Avenue	Ped2
Reconstruction	86.3%/20.1%	85.3%/19.6%
Prediction	87.5%/19.4%	96.7%/9.4%
Proposed	**87.7%/19.4%**	**97.5%/6.7%**

listed in Table D1 also proves that neither the reconstruction model nor the prediction model can reach the AUC performance achieved by combining the two models.

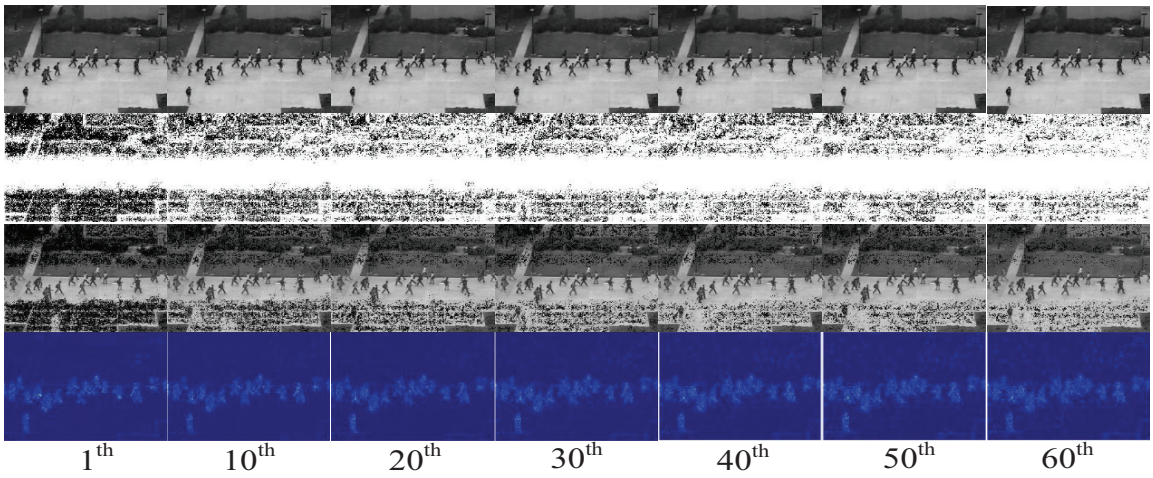


Figure D3 Visualization results related to reverse erasure under different training epochs, including: raw frames before erasing (first row), binary masks (second row), frames after erasing some pixels (third row), and reconstruction error maps (fourth row). In the error map, lighter color represents larger error.

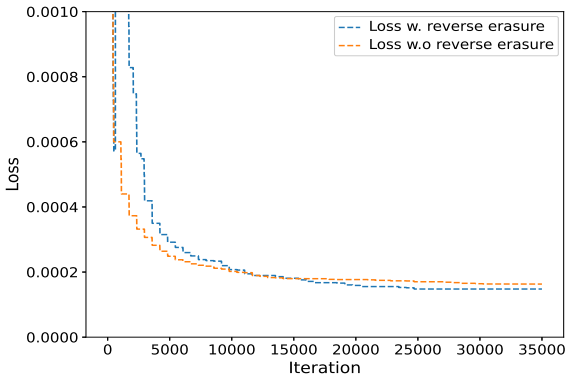


Figure D4 The training loss comparison of the proposed model with (w.) and without (w.o) reverse erasure on Ped2 dataset. The training loss is smoothed by median filter.

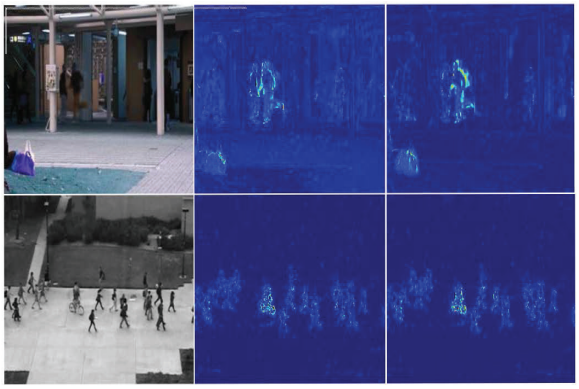


Figure D5 Visualization results of the model with and without reverse erasure on Avenue (top) and Ped2 (bottom) datasets. The images from left to right are the raw frame, the reconstruction error map without reverse erasure and the reconstruction error map with reverse erasure, respectively. Lighter color means larger error.

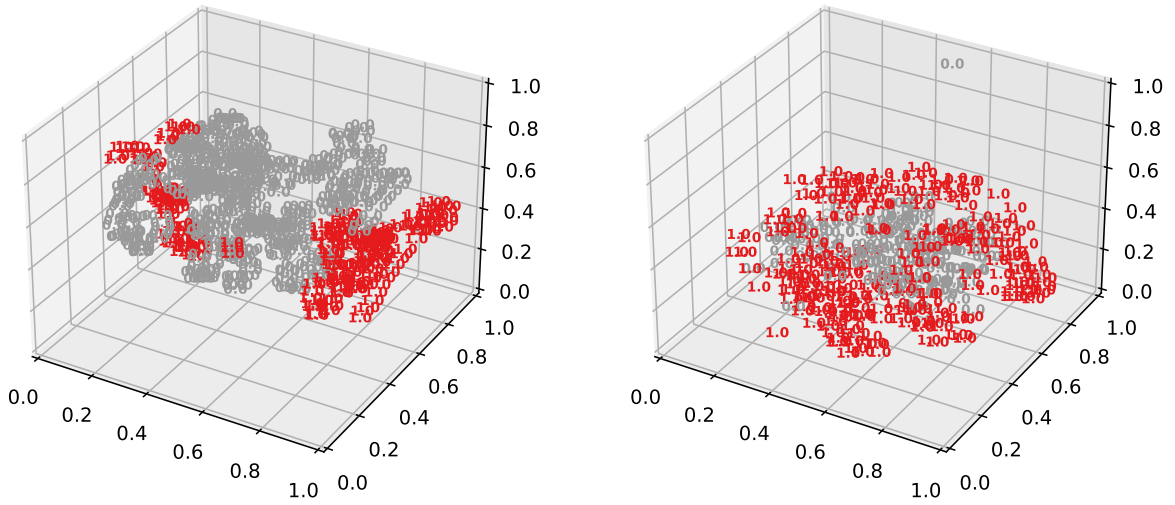
Table D2 AUC/EER comparison of the proposed model without and with reverse erasure.

	Avenue	Ped2
Without reverse erasure	86.6%/21.3%	96.8%/8.4%
With reverse erasure	**87.7%/19.4%**	**97.5%/6.7%**

Appendix D.2 Effect of reverse erasure based on reconstruction error

In this subsection, we explore the influence of reverse erasure based on reconstruction error. Figure D3 gives the masks used for erasure under different training epochs, as well as the frame images before and after erasing. From the figure, we can find that the erased pixels are mainly background pixels in each epoch, which helps the model focus more on complex foreground. And with the increase of training epoch, more background pixels are retained in the erased frame, indicating the reconstruction error gap between foreground and background is decreasing. These observations reflect that reverse erasure can effectively guide the model to reduce the reconstruction error of foreground pixels. It can also be verified in the reconstruction error maps provided in Figure D3.

To better demonstrate the advantages of reverse erasure, we conducted an ablation experiment on reverse erasure. The training loss for the proposed model with and without reverse erasure on Ped2 is plotted as the showcase in Figure D4. Although Figure D4 shows that the proposed model with reverse erasure does not significantly reduce the training loss, compared with Figure D3, we can find that the decline of the training loss is mainly dominated by foreground pixels, rather than background pixels. On the contrary, the model without reverse erasure loses guidance and treats all regions equally, resulting in the convergence of model dominated by simple background. Finally, we list the AUC performance of the model with and without reverse erasure in Table D2, and give the visual comparison in Figure D5. The results show that the model with reverse erasure gets better detection performance.



(a) t-SNE visualization on Avenue dataset

(b) t-SNE visualization on Ped2 dataset

Figure D6 t-SNE visualization of low-dimension representations of some reconstruction frames in Avenue and Ped2 datasets. The gray "0.0" represents normal, and the red "1.0" represents anomaly.

Table D3 AUC/EER comparison of the proposed model with different constraints on latent feature space.

	Avenue		Ped2	
	AUC/EER(frame)	AUC/EER(feature)	AUC/EER(frame)	AUC/EER(feature)
DESD	86.4%/20.3%	–/–	96.9%/8.5%	–/–
DE-SVDD-SD	87.3%/19.1%	53.6%/46.9%	96.4%/9.4%	56.7%/44.1%
DESD-SVDD (Proposed)	**87.7%/19.4%**	**74.1%/32.2%**	**97.5%/6.7%**	**87.6%/18.8%**

Appendix D.3 Effect of Deep SVDD

Based on t-distributed Stochastic Neighbor Embedding (t-SNE) method [7], the t-SNE visualization of the low-dimension representation of reconstruction frame on Avenue and Ped2 datasets is provided in Figure D6. We can observe that in three-dimensional space, especially in Ped2 dataset, most of normal data are gathered in the form of approaching a sphere, and abnormal data are scattered outside the sphere. This result is attributed to the feature compact loss based on Deep SVDD, which aims to find a hypersphere with minimum volume to contain normal data but not abnormal data.

In order to further demonstrate the advantages of applying Deep SVDD behind decoder, we explore three different methods:1) the mapping encoder behind decoder is removed and there are no constraints on features, which is a simple double encoding and single decoding structure, denoted as DESD; 2) Deep SVDD is performed at the bottleneck between encoder and decoder, i.e., the spatio-temporal representation of input frames are mapped into a compact hypersphere, denoted as DE-SVDD-SD; 3) the proposed method, which executes Deep SVDD behind the decoder, denoted as DESD-SVDD.

The detection performance of different methods is summarized in Table D3. In the table, AUC/EER (frame) is calculated based on reconstruction error of frame as mentioned in letter. AUC/EER (feature) is calculated according to the distance between the low-dimension feature of frame and the center of hypersphere. Firstly, the distance between features is defined as follows:

$$DIST(I_t) = \|\Phi(I_t; W^*) - c\|_2^2, \tag{D1}$$

where W^* is the parameter of a pre-training network. A large distance means that the low-dimension features of frame deviate more seriously from the normal patterns. And then the abnormality score is obtained as follows:

$$S^*_{feature}(I_t) = \frac{DIST(I_t) - min_{DIST}}{max_{DIST} - min_{DIST}}. \tag{D2}$$

From Table D3, we can observe that, regardless of AUC/EER (frame) or AUC/EER (feature), DESD-SVDD gets the better performance on both datasets. The AUC (frame) of DE-SVDD-SD is lower than that of DESD-SVDD, which confirms that, even if the high-level features are limited, the abnormal frame reconstructed by decoder may not be close to normal frame, due to the strong representation capacity of Convolutional Neural Network.

Appendix E Additional discussions

In this section, we explore the performance of DESDnet from four perspectives: 1) feature fusion methods; 2) loss functions; 3) constraints on motion; 4) abnormality score calculation methods.

Table E1 AUC/EER, space and time performance comparison of the proposed model with different feature fusion methods. Training time and inference time are counted in seconds (s).

Fusion method	Memory	Training time per epoch	Inference time per frame	AUC/EER
Concatenating	5416MB	186.3s	0.0196s	96.9%/9.4%
Convolution	5350MB	176.5s	0.0189s	97.5%/6.7%

Table E2 AUC/EER performance comparison of the proposed model with different loss function terms.

L_{int}	L_{rgb}	$L_{compact}$	Avenue	Ped2
✔	✗	✗	86.5%/20.8%	96.0%/9.1%
✔	✔	✗	87.3%/19.1%	96.9%/8.6%
✔	✗	✔	87.1%/20.1%	96.1%/9.9%
✔	✔	✔	**87.7%/19.4%**	**97.5%/6.7%**

Table E3 AUC/EER performance of the proposed model with different constraints on motion.

	Avenue	Ped2
Motion loss [8]	87.6%/19.2%	96.7%/9.0%
Weighted RGB loss (Proposed)	**87.7%/19.4%**	**97.5%/6.7%**

Table E4 AUC/EER comparison of the proposed model for different weights of the weighted RGB loss on Ped2 dataset.

λ_{rgb}	0	0.1	0.2	0.3	0.4
AUC	96.1%/8.9%	96.7%/9.4%	**97.5%/6.7%**	96.4%/9.5%	96.5%/9.1%

Appendix E.1 Comparison in terms of fusion method

In order to show the advantages of fusion module, we carried out the following experiments on Ped2 dataset to compare the convolution fusion with the conventional concatenating features method. The experimental results are listed in Table E1. We can see that, compared with the concatenating features method, there are less time and space complexity by convolution to fuse the spatial and temporal features. The AUC improved by 0.6%, and the EER effectively decreased by 2.7%.

Appendix E.2 Comparison in terms of loss functions

To evaluate the effectiveness of different loss function terms, we take L_{int} as the basic loss, and add L_{rgb} and $L_{compact}$ to it respectively to constraint the training of DESDnet. The AUC/EER performance of DESDnet with different loss items is listed in Table E2. From the table, it can be observed that L_{rgb} and $L_{compact}$ can bring certain AUC improvement respectively, while integrating both can achieve the best detection performance.

Appendix E.3 Comparisons in terms of constraints on motion

The influence of the weighted RGB loss is studied by comparing with the motion loss proposed in [8], which calculates the RGB difference between two adjacent frames. The anomaly detection results is listed in Table E3. The results show that the weighted RGB loss is able to give a higher AUC on both Ped2 and Avenue datasets.

Besides that, in experiments, we find that fixing the parameter of the weighted RGB loss λ_{rgb} as 0.2 achieves good detection performance on different datasets. We also take Ped2 dataset as an example to conduct the parameter analysis of λ_{rgb}. The experimental results are summarized in Table E4.

Appendix E.4 Comparisons in terms of abnormality score calculation

In this section, we study the impact of abnormality score calculated based on the whole frame and the patch on the detection performance. The results are listed in Table E5. From the table, it can be observed that the method of calculating abnormality score based on patch significantly improved the average abnormality score of abnormal frame, and also expand the score gap between normal and abnormal frames. The higher AUC also proves the effectiveness of the method. To accurately detect local anomalies, some methods divide the frame into multiple patches before training and train model for the patches from different regions respectively [1]. Compared with such methods, the time complexity caused by partitioning when detecting anomaly is greatly reduced.

Appendix E.5 Running time

With an NVIDIA GeForce GTX 1080 GPU, our method takes on 0.0189 seconds to evaluate abnormality for an frame of size $256 \times 256 \times 3$ on UCSD Ped2, i.e. we achieve about 53 fps for anomaly detection. With the same configuration and calculation, the MemAE in [9] averagely takes 0.0256 seconds for one frame (i.e. 39 fps), and the MNAD in [10] takes 0.0179 seconds (i.e. 56 fps). Comparing to the two AE model, it can be seen that although a prediction encoder is added to our model, the inference speed is not significantly affected because the two branches of prediction encoding and reconstruction encoding are parallel. Besides, our model avoids the computational cost of conversion in memory module.

References

1 Lu C, Shi J, Jia J. Abnormal event detection at 150 fps in matlab. In: Proceedings of 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 2013. 2720-2727

Table E5 Anomaly detection results of different abnormality score calculation methods, in which score gap represents the average score of abnormal frames minus that of normal frames.

		Avenue	Ped2
Average score of normal frames	Based on frame	0.018	0.080
	Based on patch	0.022	0.070
Average score of abnormal frames	Based on frame	0.071	0.453
	Based on patch	0.089	0.485
Score gap	Based on frame	0.053	0.382
	Based on patch	0.068	0.396
AUC/EER	Based on frame	85.5%/23.9%	95.7%/8.8%
	Based on patch	87.7%/19.4%	97.5%/6.7%

2 Mahadevan V, Li W, Bhalodia V, et al. Anomaly detection in crowded scenes. In: Proceedings of 23th IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, California, 2010. 1975-1981
3 Luo W, Liu W, Gao S. A revisit of sparse coding based anomaly detection in stacked rnn framework. In: Proceedings of IEEE International Conference on Computer Vision, Venice, Italy, 2017. 341-349
4 Paszke A, Gross S, Chintala S, et al. Automatic differentiation in PyTorch. 2017
5 Kingma D P, Ba J. Adam: a method for stochastic optimization. Computer Science, 2014
6 Loshchilov I, Hutter F. SGDR: stochastic gradient descent with warm restarts. In: Proceedings of 5th International Conference on Learning Representations, Palais des Congrés Neptune, Toulon, 2016
7 Maaten L, Hinton G. Visualizing data using t-SNE. J. Mach. Learn. Res., 2008, 9: 2579-2605
8 Li Y, Cai Y, Liu J, et al. Spatio-temporal unity networking for video anomaly detection. IEEE Access, 2019, 7: 172425-172432
9 Gong D, Liu L, Le V, et al. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In: Proceedings of IEEE International Conference on Computer Vision, Seoul, South Korea, 2019. 1705-1714
10 Park H, Noh J, Ham B. Learning memory-guided normality for anomaly detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA, 2020. 14372-14381