Journal of Computers Vol. 34 No. 5, October 2023, pp. 229-241
doi: 10.53106/199115992023103405018

Research on Video Anomaly Detection Based on Cascaded Memory-augmented Autoencoder

Lin Zhang[1], Zhao-Bo Chen[1*], Xiao-Xuan Ma[1], Fan-Bo Zhang[2], Ze-Hui Li[1], Xian-Ying Shan[1]

[1] School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture,
Beijing 102616, China

zhanglin@bucea.edu.cn, chenzhaobo97@163.com, maxiaoxuan@bucea.edu.cn,
3431396649@qq.com, 1198437590@qq.com

[2] Bank of Communications, Beijing 100031, China

Received 1 August 2023; Revised 1 September 2023; Accepted 28 September 2023

Abstract. Video anomaly detection (VAD) is crucial in public safety and intelligent video surveillance systems and has been widely researched and applied in academia. This paper proposes a video anomaly detection method based on the Cascaded Memory-augmented Autoencoder (CMAAE). CMAAE stores feature prototypes of standard samples in a memory pool and embed multiple memory-enhancing modules in the encoder-decoder structure. SE attention is introduced into memory modules to improve their performance, and skip connections are used to share attention weights among memory modules, enabling the model to learn more comprehensive feature information and enhance the quality of reconstructed video frames. Multiple loss constraint models are used during training to improve anomaly detection accuracy. CMAAE achieves outstanding performance of 99.2% on the UCSD Ped2 dataset and 89.4% on the CUHK Avenue dataset, demonstrating the effectiveness of our approach.

Keywords: video anomaly detection, memory network, autoencoder, abnormal events

1 Introduction

In recent years, to improve the safety of the public environment, public surveillance systems have been deployed on a large scale, generating a large amount of surveillance video data in real-time and increasing the cost of surveillance. In order to save labor costs and improve surveillance capability, we urgently need intelligent surveillance systems to realize the automatic detection of anomalies. This problem has attracted extensive attention from the academic community; many scholars have invested in the research and have many research results [1-3]. However, video anomaly detection poses significant challenges due to the environment-dependent nature of anomalous events and their infrequent occurrence. Collecting enough anomaly samples to model all possible anomalous events comprehensively is often challenging. As a result, a typical approach in video anomaly detection is to train an unsupervised learning model on a dataset of normal events to learn their representative patterns. Subsequently, during testing, samples that deviate significantly from the learned normal patterns are classified as anomalous [4-5].

With the rapid development of deep learning, deep learning has been widely applied in various tasks and achieved remarkable success. Among the deep learning-based video anomaly detection methods, a common approach is constructing a reconstruction model based on an autoencoder (AE), which aims to reconstruct the input and recognize anomalies with significant reconstruction errors [6]. These methods are implemented because normal frames have minor reconstruction errors, and abnormal frames have significant reconstruction errors to detect anomalies. Existing deep learning methods usually use complex networks to represent video data features. However, these networks are always accompanied by generalization to unknown samples and can reconstruct anomalous data well. As a result, network models trained to distinguish anomalous data by the size of reconstruction error are not guaranteed to detect anomalous data in the testing phase [7]. To mitigate this shortcoming, recent studies [5-6] suggest adding memory module at the AE network to constrain the model's generalization ability. This memory module serves the purpose of recording normal data features during the training and testing phases. During reconstruction, the model matches the input features with the stored normal data features within

[*] Corresponding Author

the memory module. This matching procedure amplifies the reconstruction error for abnormal samples, effectively mitigating the shortcomings and improving the model's ability to detect anomalies.

Inspired by previous approaches, we propose a deep learning model for video anomaly detection based on a cascaded memory-augmented autoencoder (CMAAE) video anomaly detection task. The model embeds three memory-enhanced modules in the bottleneck and up-sampling process, while skip-connections are made between these three memory modules to transfer the attention weights generated by the modules to learn deeper video frame data features. The SE attention mechanism and the soft shrinkage sparse function are added to the memory modules to enhance their performance of the memory modules, allowing them to learn to memorize more critical features during the training process. The model also uses skip-connections to utilize the encoder's low-level and high-level features during the up-sampling process; this helps to improve the quality of the reconstruction. Multiple loss functions train the model to compare the differences between the reconstructed and original video frames, allowing the model to detect anomalies better.

In summary, this paper makes the following key contributions:

(1) It embeds multiple memory modules in the bottleneck and up-sampling processes of an encoder-decoder structure with skip connections while achieving the sharing of attention weights within the memory modules. This ensures better capturing of expected features, thereby sensitively identifying anomalous events. Multiple loss functions also train the model, enhancing anomaly detection accuracy.

(2) Improvement of the memory modules. Introducing SE attention mechanisms and soft-shrinkage sparse functions in the memory modules enhances the module's ability to learn normal memory features.

(3) Extensive experiments on the UCSD Ped2 and CUHK Avenue, demonstrate the effectiveness of our approach.

The remaining sections of this paper are organized as follows: Section 2 provides an overview of recent research on video anomaly detection. In Section 3, we introduce our Cascade Memory-augmented Autoencoder method. Section 4 presents our experimental results, and Section 5 concludes the paper.

2 Related Work

2.1 Anormal Detection

Video anomaly detection is usually treated as an unsupervised learning problem due to the paucity of data and difficulty in obtaining annotations, using only normal data for model training. Over the past few years, CNN-based approaches have witnessed notable advancements in anomaly detection. Numerous methods for anomaly detection leverage AE-based reconstructions as a means to represent features, such as Conv- AE [4], Conv-LSTM [8], etc. These methods also need some help. They also tend to reconstruct anomalous features, which decreases the detection accuracy. Other methods have been investigated to address this problem. For example, Liu et al. [7] proposed to predict future images and use the prediction error as training loss. Yu et al. [9] used GANs to learn the normality of the data and proposed Adversarial Event Prediction (AEP) networks to suppress learning of representations of past events and force learning to predict future events to explore their correlations. Zhao et al. [10] used spatiotemporal LSTM (ST-LSTM) networks to extract and store appearance and motion changes. Inspired by GANs, they introduced a discriminator into adversarial training with the ST-LSTM to improve learning spatiotemporal correlations between continuous video images.

2.2 Memory Module

The distinction between continuous video images and still images lies in the presence of temporal changes, as objects within a video exhibit ongoing and evolving patterns. Modern networks often incorporate memory functions such as internal memory units or attention mechanisms to capture the interactions between consecutive frames. These memory functions enable the network to retain and utilize information from previous frames when processing subsequent frames. Graves et al. [11] combined a feature extraction neural network with a dynamic external memory module to improve its computational and storage capabilities. Yu et al. [12] introduced a novel long-term segmentation tracker (LTST) incorporating a memory attention network. This approach enables online learning and addresses the challenge of limited long-term adaptation in video object segmentation. However, for video anomaly detection, the memory capacity of this method may need to be enhanced to accommodate all the necessary information.

Similarly, Fernando et al. [13] proposed a plastic neural memory access mechanism for video anomaly detection. This mechanism utilizes memory reading and writing operations and generates static and dynamic connection weights to enhance learning. Additionally, Gong et al. [5] extended a deep autoencoder by incorporating an external memory module. They compared it to a pure deep autoencoder, which significantly improved the video anomaly detection performance compared to the pure deep autoencoder. Subsequently, Park et al. [6] improved the update strategy of the memory module. In these methods, the memory module is placed at the middle layer of the AE. Liu et al. [14] used optical flow to train optical flow reconstruction networks with multiple memory modules. Hou et al. [15] proposed a multi-scale memory mechanism that divides the feature map into multiple blocks and assembles the blocks as needed, which affects the quality of image reconstruction and helps distinguish between normal and abnormal samples.

3 Proposed Abnormal Detection

In this paper, the Cascaded Memory-augmented Autoencoder (CMAAE) model uses an autoencoder widely used in recent years based on the U-net architecture. As shown in Fig. 1, the cascaded memory-augmented autoencoder (CMAAE) consists of an encoder, a decoder, and a memory module. The encoder extracts the deep features contained in the video by stepwise down-sampling based on the input video sequence. The memory module serves as a repository for storing the normal patterns acquired by the model during the training phase. Subsequently, when the model processes query features extracted by the encoder, it accesses the relevant memory slots within the memory module to assess the similarity between the encoded features and the stored patterns. The output of the memory module, along with the query features, is aggregated and passed as input to each layer of the decoder. Ultimately, the decoder utilizes the aggregated features to reconstruct the video frames of the input video sequence.

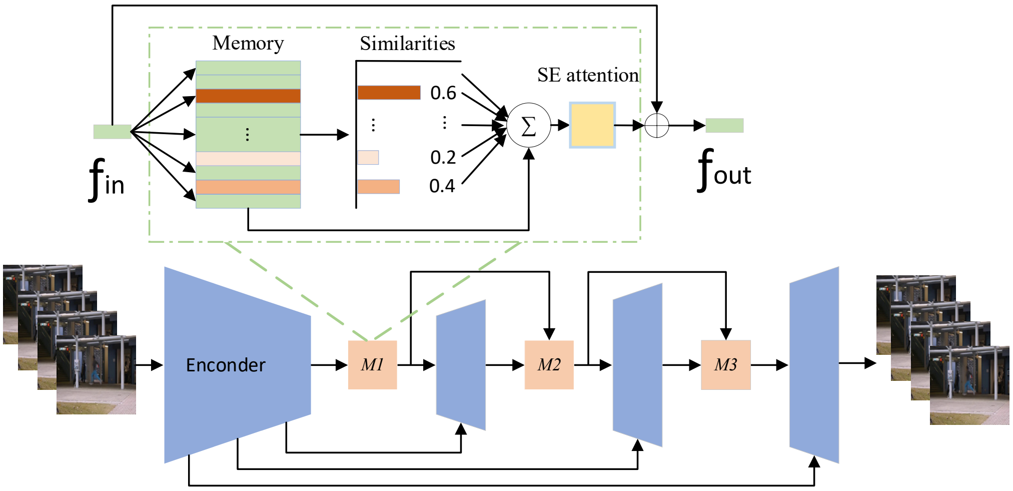


Fig. 1. Structure of the Cascade Memory-Augmentation Autoencoder (CMAAE)

(It consists of three memory augmented modules with SE attention mechanisms and skip-connected autoencoders and uses weight connections between memory modules.)

3.1 Memory Networks

To enable the model to capture feature-rich representations of normal data as well as more critical data features, we use an implementation similar to [5], where the memory module consists mainly of memory matrices, SE attention blocks [16], and soft shrinkage sparse blocks [17]. The memory module is implemented as a matrix $M \in R^{N \times C}$, consisting of N memory entries of fixed dimension C. Each row of the matrix M is called memory entry m$_i$, where i =1, 2, 3, ..., N. The primary objective of the memory module is to represent the input features using a weighted sum of similar memory entries, enabling it to recall normal patterns effectively. The realization of the query feature z_i is input to the memory module to find the similar feature in the memory entry m_i with the relationship shown below:

$$\hat{z} = \omega M = \sum_{i=}^{N} \omega_i m_i \tag{1}$$

Where the maximum capacity of the memory module is defined by the hyperparameter N, ω_i denotes the memory element m_i, and the query feature z_i to calculate the similarity weight, which is calculated as follows:

$$\omega_i = \frac{e^{D_{COS}(z_i, m_i}) }{\sum_{j=1}^{N} e^{D_{COS}(z_i, m_j)}}. \tag{2}$$

D_{COS} denotes the cosine similarity between each query z_i and each memory item m_j:

$$D_{COS}(z_i, m_i) = \frac{z_i m_i^T}{\|z_i\| \|m_i\|}. \tag{3}$$

When utilizing a limited number of normal patterns within the memory matrix for reconstruction, which favors significant reconstruction errors in the reconstruction of abnormal data, a soft shrinkage operation is used to promote sparsity of similar attention weights ω:

$$\hat{\omega}_i = R_{soft}(\omega_i, \lambda) = \begin{cases} \omega_i + \lambda & \omega_i \le -\lambda, \\ 0 & |\omega_i| \le \lambda \\ \omega_i - \lambda & \omega_i \ge \lambda. \end{cases}, \tag{4}$$

Where $\hat{\omega}_i$ represents the i-th weight of the memory-addressed attentional weight vector influenced by memory and has undergone soft contraction, while λ denotes the threshold for the contraction process. Since ω are non-negative values, soft shrinkage is transformed using the continuous ReLU activation function.

The model is encouraged to represent each example with a smaller but more relevant memory content by employing sparse addressing. This approach enables the model to learn more informative representations in memory, enhancing its ability to capture important features and patterns in the data. Similar to sparse representations [18], promoting sparsity in addressing weights facilitates the testing process by training memory M to account for ω sparsity. Promoting sparsity in ω also mitigates the problem of anomalous samples that are well reconstructed with dense addressing weights.

Applying the SE attention module after matching to the most similar features after a memory matrix can adaptively recalibrate the channel dependencies in the feature map, emphasize the useful features in it and suppress the useless ones to obtain new features that are more effective. Fig. 2 shows the structure of the SE attention model.

$$\hat{Z} = f_{se}(\hat{z}) * z. \tag{5}$$

Three memory modules, M1, M2, and M3, are embedded in the bottleneck and uptake process of the cascade memory-augmented autoencoder (CMAAE) model. In previous work [14], three memory modules are simply embedded without circulation of attentional weights, and the memory modules learn a large amount of similar knowledge repetitively, which reduces the joint expressive power of the model. Therefore, we added the connection of attentional weights between the memory modules.

$$\omega_{mem2} = F_{mem2}(\hat{Z}_1) + \omega_{mem1}. \quad \textbf{(6)}$$

Where ω_{mem2} denotes the attentional weight output by M2, ω_{mem1} denotes the attentional weight output by M1, and $F_{mem2}(\hat{Z}_1)$ denotes the attentional weight obtained by inputting the features output by M1 into M2 after up-sampling.

$$\omega_{mem3} = F_{mem3}(\hat{Z}_2) + \omega_{mem2}. \quad \textbf{(7)}$$

Where ω_{mem3} denotes the attentional weight of the M3 output, which is computed similarly to ω_{mem2}.

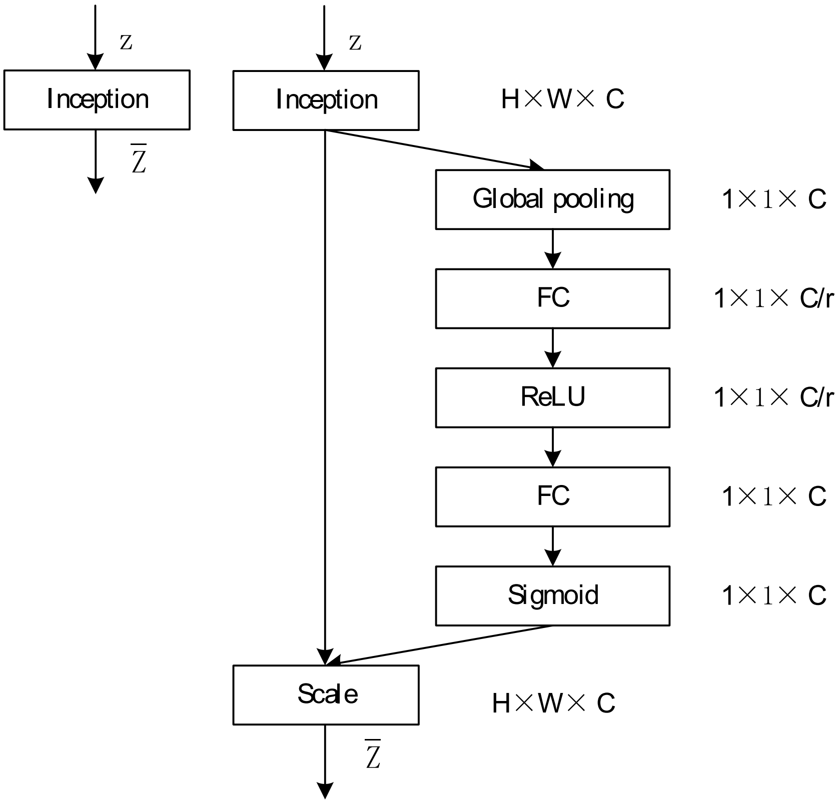


Fig. 2. The SE attention module structure

3.2 Train Loss

During the training of CMAAE, normal data is fed into the model to reconstruct the input data. Denoting the input data as y and the reconstruction result as y', the model utilizes the L2 paradigm as the reconstruction loss function. This loss function penalizes the distance between the input image y and the reconstructed image y', ensuring accurate reconstruction.:

$$\mathcal{L}_{recon} = \|y - y'\|_2^2. \quad \textbf{(8)}$$

For each memory module, we introduce the entropy loss to the matching probabilities as follows:

$$\mathcal{L}_{mem} = \sum_{i=1}^{S} \sum_{j=1}^{N} -\omega_{i,j} \log(\omega_{i,j}). \quad \textbf{(9)}$$

Where S represent the number of memory modules and $\omega_{i,j}$ denotes the matching probabilities.

Since the memory network is updated at each iteration and retaining too much previously learned information is challenging, transient loss [19] prevents frequent memory changes and balances memory stability and learning capacity. It facilitates the learning of contextual spatial-temporal relationships in video sequences.

$$\mathcal{L}_{tran} = \sum \|M - M'\|_2^2. \quad \textbf{(10)}$$

Where M denotes the memory matrix in each memory module, and M' denotes the transient value of the memory matrix in the previous training step, the difference between the two matrices is computed using the L2 paradigm, and then the sum of the transient losses of all memory modules is obtained.

Due to the problem of shooting angle and distance, the farther the distance, the more blurred things are in detail, and the abnormal video sequences, there are some abnormalities with little difference from the normal events. Therefore, the introduction of Structural Similarity Loss (SSIM) [20] to train the model helps the model learn the detailed information of reconstructed video frames, consider pixel-level differences, and enhance the quality of reconstructed images. Distinguishing the subtle differences between abnormal and normal events and increasing the reconstruction error of abnormal frames benefit the model's performance.

$$\mathcal{L}_{ssim} = f_{ssim}(y, y') = \frac{(2\mu_y \mu_{y'} + C_1)(2\delta_{yy'} + C_2)}{(\mu_y^2 \mu_{y'}^2 + C_1)(\delta_y^2 \delta_{y'}^2 + C_2)}. \quad \textbf{(11)}$$

Where μ_y and $\mu_{y'}$ represent the mean of the input and output, respectively; δ_y^2 and $\delta_{y'}^2$ denote the variance of the input and output, respectively; and $\delta_{yy'}$ denotes the covariance of the input and output, with C_1 and C_2 as constants.

In order to train the CMAAE model, we set the corresponding hyperparameters λ_{recon} , λ_{mem} , λ_{tran} and λ_{ssim} to achieve a balance among the above loss functions. As a result, we obtain the following set of loss functions for training the model:

$$\mathcal{L}_{CMAAE} = \lambda_{recon}\mathcal{L}_{recon} + \lambda_{mem}\mathcal{L}_{mem} + \lambda_{tran}\mathcal{L}_{tran} + \lambda_{ssim}\mathcal{L}_{ssim}. \quad \textbf{(12)}$$

4 Experiments and Analysis

4.1 Datasets

To evaluate the performance of the proposed CMAAE and compare it with state-of-the-art algorithms, we perform experimental validation on two public dataset benchmarks, the UCSD Ped2 dataset [21] and the CUHK Avenue dataset [22]. Fig. 3 depicts the abnormal and normal events on the two datasets, respectively. The training dataset comprises standard samples, while the test dataset comprises normal and abnormal samples.

The UCSD Ped2 dataset [21] consists of 16 training videos and 12 test videos with a resolution of 360×240, acquired with a fixed camera. Crowd density on the sidewalk fluctuates frequently, from sparse to very crowded.

The training set contains only normal data, i.e. pedestrians walking, and the test set contains normal and abnormal situations. The abnormal situations contain intrusion of non-human entities such as car intrusion and abnormal movement patterns of pedestrians such as skateboards and bicycles.

Lu et al. [22] introduced their anomaly detection method and generated the CUHK Avenue dataset. The dataset consists of 16 training and 21 test videos, capturing 47 anomalous events. The dataset comprises 30,652 frames, with 15,328 frames allocated for training and 15,324 frames for testing purposes. The dataset includes abnormal activities such as throwing objects, wandering, and running. It is worth noting that the person's size may vary depending on the camera's position and angle. Training videos contain videos of normal situations. The test video contains normal and abnormal event videos. Furthermore, each frame has pixel-level labeling.

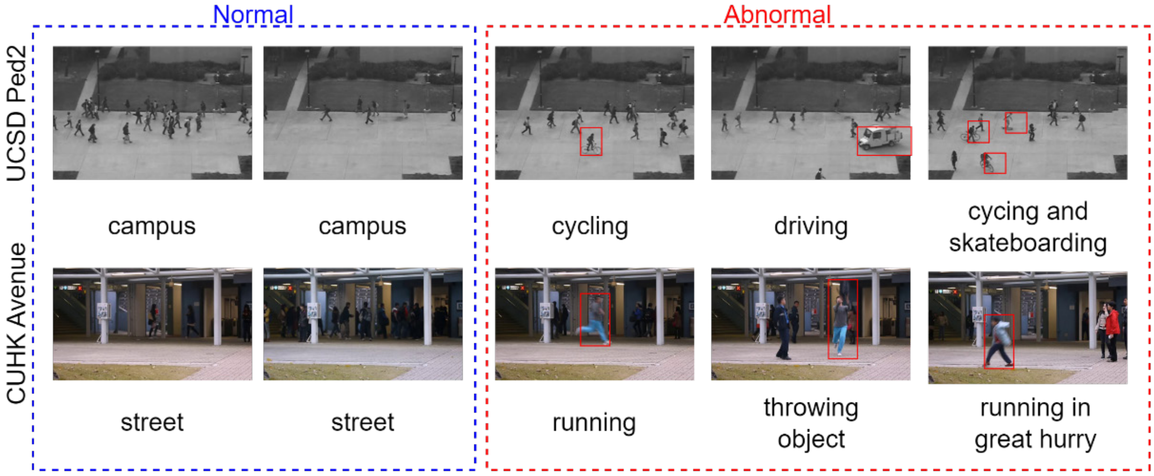


Fig. 3. Illustration of dataset types

(We list the different abnormal behaviors of the two datasets, with those identified as abnormal highlighted in red.)

4.2 Evaluation Criterion

The thresholds for changing the anomaly scores and calculating the actual positive rate (TPR) and false positive rate (FPR), commonly used in video anomaly detection tasks, are used in the experiments to generate the ROC curves. As in most previous work, the area under the ROC curve at the frame level, AUC, is used to evaluate the model's performance. It effectively avoids the subjectivity of parameter setting. It makes the experimental results more reasonable, which is especially suitable for evaluating the performance of the detection task when the data distribution is highly unbalanced. In the anomaly detection task, the higher the value of AUC, the better the performance of the algorithm detection.

4.3 Implementation Details

The RTX3090 graphics card and Pytorch framework were used to construct the model in the experiments. Preprocess the raw data in all datasets, resize the video frames to 32×32, and standardize them; set the capacity of each memory module to 2K with a shrinkage threshold of 5e-4. The Adam optimizer [17] was used for optimization, with the learning rate initialized to 0.01 and decayed by 0.8 every 20 epochs. The batch sizes of Ped2 and Avenue were 256, the epoch was set to 100, and the training loss-weight hyperparameters λ_{recon} , λ_{mem} , λ_{tran} , and λ_{ssim} were set to 1.0, 2e^{-4}, 0.1, and 2e^{-4}, respectively.

4.4 Result

In order to illustrate the improvement in the detection performance of the anomaly detection task due to the model CMAAE, we compare the method in this paper with other state-of-the-art anomaly detection methods in recent years, such as MemAE [5], MNAD [6], DAML [24], AMAE [29], and MGAE [27]. We obtain performance data for each method on the Ped2 and Avenue datasets from the original paper. The comparison results are shown in Table 1.

Table 1. Comparison of experimental results with other state-of-the-art methods

Year	Methods	AUC (%)	
		Ped2	Avenue
2019	MemAE [5]	94.1	83.3
2020	MNAD [6]	97.0	88.5
2020	IPR [25]	96.2	83.7
2020	CDAE [26]	96.5	86.0
2021	DAML [24]	95.1	88.8
2021	HF^2-VAD w/o FP [14]	98.8	86.8
2022	STCEN [28]	96.9	86.6
2022	AMAE [29]	97.4	88.2
2022	CRVAD-GAN [30]	96.3	87.1
2022	MTT [31]	97.8	88.5
2023	MGAE [27]	97.8	87.6
2023	ARAE [23]	97.4	86.7
2023	CMAAE (our)	99.2	89.4

As shown in Table 1, CMAAE achieves better results than the state-of-the-art methods in recent years on both UCSD Ped2 and CUHK Avenue datasets, proving our method's effectiveness. Among the reconstruction anomaly detection methods, the CMAAE model obtains an accuracy of 99.2% on the UCSD Ped2 dataset, which is also a 0.4% accuracy improvement over the previous best-performing HF2-VAD w/o FP [14]. However, it also uses three memory modules embedded in the model. The contents of these memory modules learn a large amount of redundancy in isolation. In contrast, the CMAAE model carries out a flow of attention weights of memory modules between the embedded memory modules, which promotes the model to learn more comprehensive global information and improves the model detection accuracy. An accuracy of 89.4% was obtained on the CUHK Avenue dataset, which is 0.6% higher than the previous best method, DAML [24], whose model learned appearance and motion separately, ignoring the correlation between appearance and motion information, resulting in incomplete information being learned. In contrast, CMAAE incorporates the SE attention module into the memory module to learn to remember more comprehensive data information, which enhances the robustness of the model and is more conducive to distinguishing between normal and abnormal events.

4.5 Analysis of Experimental Results

The anomalies of all frames in the video are plotted as continuous anomaly curves to demonstrate the method's effectiveness, we show in Fig. 4 and Fig. 5, respectively, the anomaly score variation curves in Ped2 and Avenue datasets, where anomalous and normal events can be accurately detected and differentiated by the level of the anomaly score curves.

Fig. 4 shows the evolution of the anomaly score curves in the Ped2 test datasets. The anomaly scores are all in the state of shallow scores on the sidewalk without anomalous events. Moreover, the curve rises rapidly when anomalous events (cycling, skateboarding) occur, indicating that this paper's method can detect anomalously. However, due to the shooting angle, the anomalies will be occluded, and the value of the obtained anomaly curve will decrease after the anomaly is occluded. Subsequently, with the occluded anomalies separated, the model detects the anomalies more accurately, and higher anomaly scores are obtained. A higher value of the anomaly curve indicates a greater likelihood of an anomalous situation. Fig. 5 shows the visualization results of the detection performed on the Avenue test dataset. The figure (left) shows a person performing a fast run back and forth twice. The anomaly score rises rapidly when the running anomaly is detected and drops rapidly to a shallow level after he disappears from the video. The figure (right) demonstrates the anomaly of a small child running. A high anomaly score is detected when he moves with a large amplitude, and the value of the anomaly curve drops rapidly to 0 after he runs out of the frame, indicating that there is no anomaly at the moment, proving the excellent detection performance and robustness of the model.

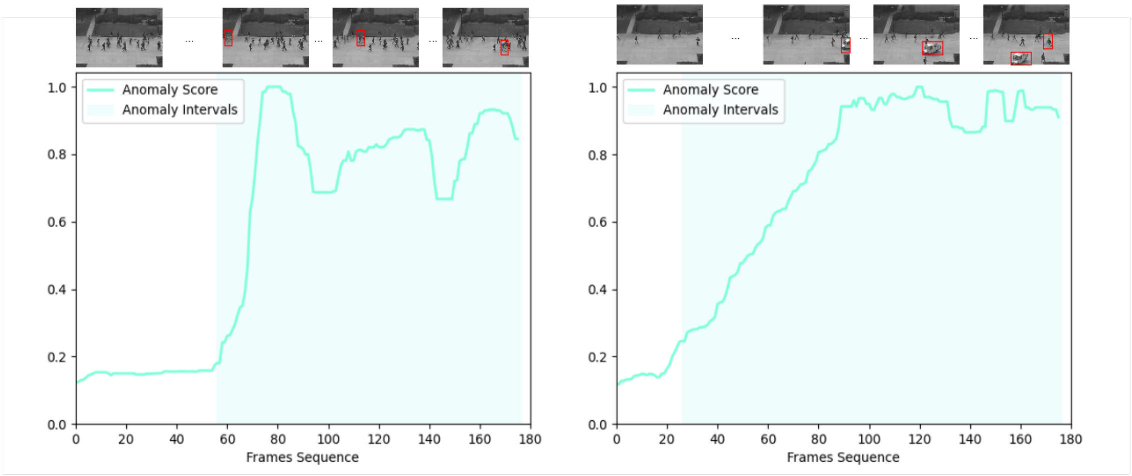


Fig. 4. Anomaly score curves of the model on the UCSD Ped2 dataset and their corresponding anomalies

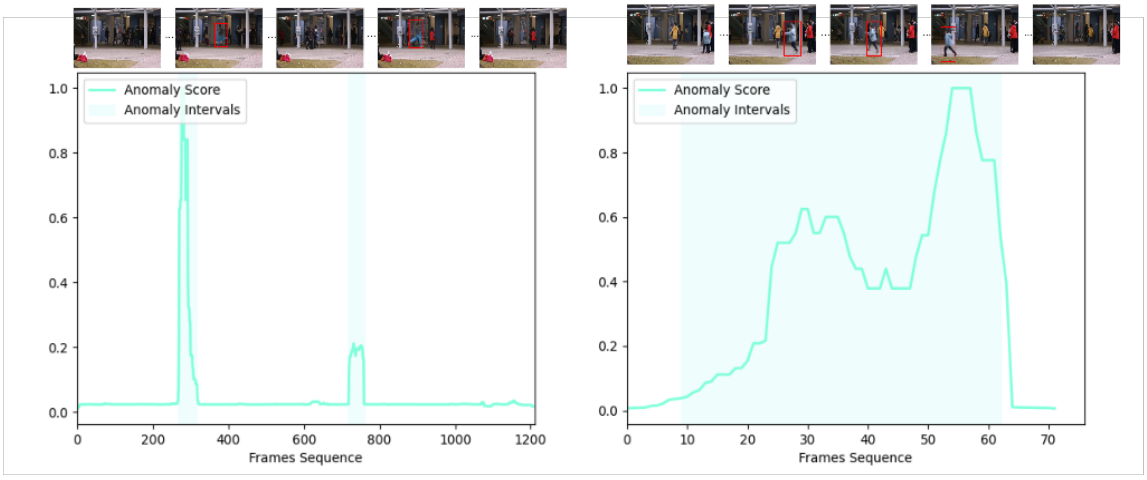


Fig. 5. Anomaly score curves of the model on the CUHK Avenue dataset and their corresponding anomalies

In Fig. 6, we show the process of accuracy AUC worth changing when the model is trained on different datasets to demonstrate the applicability of the model. During training, the model within 60 to 70 epochs has the highest AUC. In contrast, the model has a relatively low AUC in other parts of the model, indicating the applicability of the model.

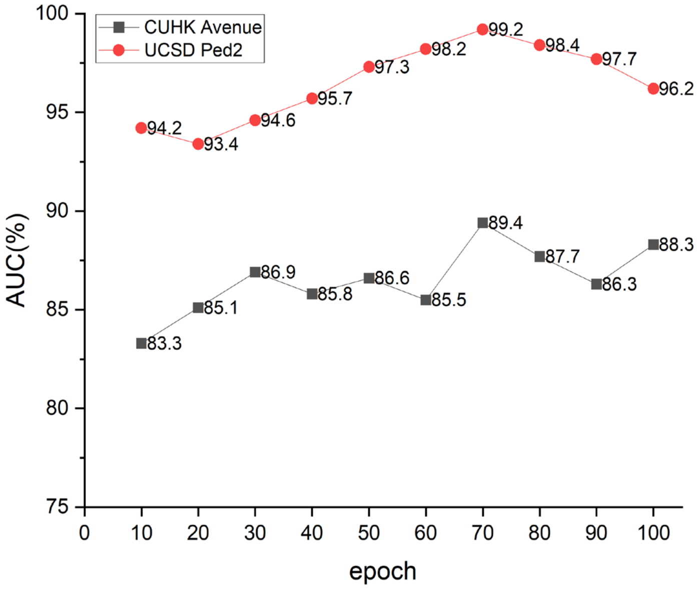


Fig. 6. Comparison of AUC for different models

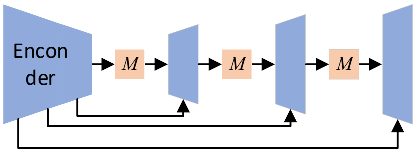
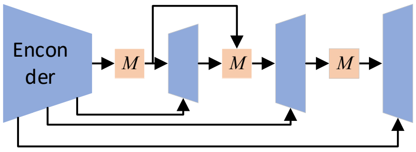
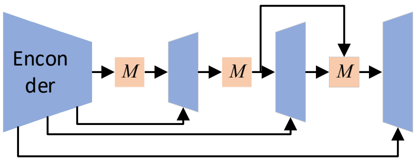
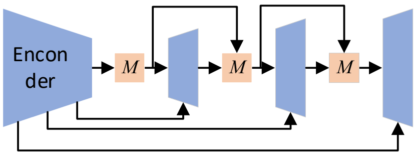
(The epoch of completion of training is used as the x-axis to show the change of AUC values during the model's training process on Ped2 and Avenue datasets.)

4.6 Ablation Studies

To thoroughly analyze the impact of various components of our model on the Ped2 dataset, we conducted a comprehensive ablation study. This study focused on assessing the effects of different components, including the weight connection of the memory module, the SE attention module, the reconstruction loss, the sparsity loss, the transient loss, and the structural similarity loss. By systematically evaluating these components, we gained valuable insights into their contributions to the overall performance of our model.

Weighted Connection Analysis. Weight connection can help the model share feature information between different levels of features, improving the model's performance. In order to demonstrate the impact of weight connections within the model's memory modules, we conducted experiments with different weight configurations. Specifically, we explored various weight connections for the three memory modules. The experimental outcomes are summarized in Table 2. The results show that after adding weight connections, the AUC value of the model increases from 98.6% to 99.2%, and the model's accuracy is significantly improved. Meanwhile, the AUC of using only the single-stage weight connection methods b and c is improved to 99.1% and 98.8%, respectively, while the improvement effect of using the two-stage weight connection method d is more pronounced, reaching 99.2%. The weights generated by the memory module connected by way b obtain higher boosts than c. This is because the attention weights generated by the memory module M1, which is at the bottleneck, record deeper-level features, while the memory module, during the up-sampling process, records relatively shallow-level features, resulting in the model presenting different performances. The dual-level connection of d can constitute the circulation of deep-level and shallow-level features, which promotes the model to learn different levels of feature information and record more comprehensive feature information, making the model more expressive.

Table 2. Comparison of results of ablation experiments on weighted connections of memory models

Index	Weighted connection	AUC (%)
a		98.6
b		99.1
c		98.8
d		99.2

Attention and Loss Function Analysis. We conducted an ablation study of the attention and loss functions of the CMAAE model on the Ped2 dataset, and the experimental results are shown in Table 3. We gradually removed the training loss and attention modules during the experiment to see the effects of SE attention and individual losses on the model performance. The results show that with the addition of SE attention, the importance weighting of the features in the memory module makes the model pay more attention to the critical features in recording normal event features, and the AUC value is improved by 1.6%. The model's performance improved by 2.2% to 97.5% after using the reconfiguration loss $\mathcal{L}_{recon}$. Adding memory sparsity loss $\mathcal{L}_{mem}$ and transient loss $\mathcal{L}_{tran}$ helps the model to manage the memory capacity and focus on storing the information that is most important for anomaly detection, which brings 0.7% and 0.4% performance improvement respectively. The structural similarity loss, on the other hand, helps the model retain the global features of the input data and promotes better anomaly capture by the model, which also brings a 0.6% improvement to the model. After weighted fusion training of these losses, the optimal model with 99.2% performance is obtained.

Table 3. Comparison of the results of ablation experiments on various loss and SE attention modules of the models

Model	SE	$\mathcal{L}_{recon}$	$\mathcal{L}_{mem}$	$\mathcal{L}_{tran}$	$\mathcal{L}_{ssim}$	AUC (%)
1						93.7
2	√					95.3
3	√	√				97.5
4	√	√	√			98.2
5	√	√	√	√		98.6
6	√	√	√	√	√	99.2

5 Conclusion

This paper proposes a deep learning model based on Cascaded Memory-augmented Autoencoder (CMAAE) for video anomaly detection. By embedding multiple memory modules in the bottleneck of the encoder-decoder structure and the up-sampling process and sharing the transfer of attention weights within the memory modules, the model improves the recording of normal features and the sensitivity to anomalous events. In addition, introducing the SE attention mechanism and the soft-shrinkage sparse function improves the performance of the memory modules, allowing them to learn better and memorize normal features. The utilization of skip connections in the up-sampling process proves beneficial as it incorporates both low-level and high-level features, resulting in improved reconstruction quality. Additionally, employing multiple loss functions aids the model in detecting anomalies more effectively. Our method demonstrates its effectiveness in anomaly detection through extensive experiments conducted on two widely used datasets, UCSD Ped2 and CUHK Avenue.

6 Acknowledgement

This research was supported by the Project Fund Research on the Analysis of Group Abnormal Behavior in a High-Density Crowd Environment Based on Social Force Model (No. 00331615022).

References

[1] Y. Liu, D. Yang, Y. Wang, J. Liu, L. Song, Generalized video anomaly event detection: Systematic taxonomy and comparison of deep models. <https://arxiv.org/abs/2302.05087>, 2023 (accessed 13.09.23).

[2] J. Ren, F. Xia, Y. Liu, I. Lee, Deep video anomaly detection: Opportunities and challenges, in: Proc. International Conference on Data Mining Workshops (ICDMW), 2021.

[3] T.-M. Tran, T.-N. Vu, N.-D. Vo, T.-V. Nguyen, K.-H. Nguyen, Anomaly analysis in images and videos: A comprehensive review, ACM Computing Surveys 55(7)(2022) 1-37.

[4] M. Hasan, J. Choi, J. Neumann, A.-K. Roy-Chowdhury, L.-S. Davis, Learning temporal regularity in video sequences, in: Proc. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[5] D. Gong, L. Liu, V. Le, B. Saha, M.-R. Mansour, S. Venkatesh, A.-V. Hengel, Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, in: Proc. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[6] H. Park, J. Noh, B. Ham, Learning memory-guided normality for anomaly detection, in: Proc. 2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[7] W. Liu, W. Luo, D. Lian, S. Gao, Future frame prediction for anomaly detection–a new baseline, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] W. Luo, W. Liu, S. Gao. Remembering history with convolutional lstm for anomaly detection, IEEE International Conference on Multimedia and Expo (ICME), 2017.

[9] K. Deepak, S. Chandrakala, C.-K. Mohan, Residual spatiotemporal autoencoder for unsupervised video anomaly detection, Signal, Image and Video Processing15(1)(2021) 215-222.

[10] J. Yu, Y. Lee, K.-C. Yow, M. Jeon, W. Pedrycz, Abnormal event detection and localization via adversarial event prediction, IEEE Transactions on Neural Networks and Learning Systems33(8)(2021) 3572-3586.

[11] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, S.-G. Colmenarejo, E. Grefenstette, T. Ramalho, J. Agapiou, A.-P. Badia, K.-M. Hermann, Y. Zwols, G. Ostrovski, A. Cain, H. King, C. Summerfield, Phil Blunsom, K. Kavukcuoglu, D. Hassabis, Hybrid computing using a neural network with dynamic external memory, Nature 538(7626)(2016) 471-476.

[12] L. Yu, B. Qiao, H. Zhang, J. Yu, X. He, LTST: Long-term segmentation tracker with memory attention network, Image and Vision Computing 119(3)(2022) 104374.

[13] T. Fernando, S. Denman, D. Ahmedt-Aristizabal, S. Sridharan, K.-R. Laurens, P. Johnston, C. Fookes, Neural memory plasticity for medical anomaly detection, Neural Networks 127(7)(2020) 67-81.

[14] Z. Liu, Y. Nie, C. Long, Q. Zhang, G. Li, A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction, in: Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[15] J. Hou, Y. Zhang, Q. Zhong, D. Xie, S. Pu, H. Zhou, Divide-and-assemble: Learning block-wise memory for unsupervised anomaly detection, in: Proc. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021.

[16] J. Hu, L. Shen, G. Sun. Squeeze-and-excitation networks, in: Proc. 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[17] D.-P. Kingma, J. Ba, Adam: A method for stochastic optimization. <https://arxiv.org/abs/1412.6980>, 2014 (accessed 14.09.23).

[18] B. Zhao, L. Fei-Fei, E.-P. Xing. Online detection of unusual events in videos via dynamic sparse coding, in: Proc. 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[19] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, Meta-learning with memory-augmented neural networks, in: Proc. 2016 International Conference on Machine Learning, PMLR 48(2016) 1842-1850.

[20] Z. Wang, A.-C. Bovik, H.-R. Sheikh, E.-P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing13(4)(2004) 600-612.

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

[22] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 fps in matlab, in: Proc. 2013 IEEE International Conference on Computer Vision (ICCV), 2013.

[23] V.-T. Le, Y.-G. Kim, Attention-based residual autoencoder for video anomaly detection, Applied Intelligence 53(3) (2023) 3240-3254.

[24] B. Li, S. Leroux, P. Simoens, Decoupled appearance and motion learning for efficient anomaly detection in surveillance video, Computer Vision and Image Understanding 210(9)(2021) 103249.

[25] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, J. Yang, Integrating prediction and reconstruction for anomaly detection, Pattern Recognition Letters 129(1)(2020) 123-130.

[26] Y. Chang, Z. Tu, W. Xie, J. Yuan, Clustering driven deep autoencoder for video anomaly detection, in: Proc. European Conference on Computer Vision, ECCV 2020, 2020.

[27] L. Zhou, J. Yang, Video anomaly detection with memory-guided multilevel embedding, International Journal of Multimedia Information Retrieval 12(1)(2023) 6.

[28] Y. Hao, J. Li, N. Wang, X. Wang, X. Gao, Spatiotemporal consistency-enhanced network for video anomaly detection, Pattern Recognition 121(1)(2022) 108232.

[29] Y. Liu, J. Liu, J. Lin, M. Zhao, L. Song, Appearance-motion united auto-encoder framework for video anomaly detection, IEEE Transactions on Circuits and Systems II: Express Briefs 69(5)(2022) 2498-2502.

[30] D. Li, X. Nie, X. Li, Y. Zhang, Y. Yin, Context-related video anomaly detection via generative adversarial network, Pattern Recognition Letters 156(4)(2022) 183-189.

[31] Y. Li, X. Song, T. Xu, Z. Feng, Memory-Token Transformer for Unsupervised Video Anomaly Detection, in: Proc. 2022 26th International Conference on Pattern Recognition (ICPR), 2022.