Image and Vision Computing 146 (2024) 105011

Contents lists available at ScienceDirect

Image and Vision Computing

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

Video anomaly detection based on a multi-layer reconstruction autoencoder with a variance attention strategy

Shifeng Li [a,*], Yan Cheng [a], Liang Zhang [a], Xi Luo [a], Ruixuan Zhang [b]

[a] School of Information Science and Technology, Bohai University, Jinzhou 121000, Liaoning, China
[b] Hikvision Research Institute, Hikvision Digital Technology Co., Ltd, Hangzhou, 310051 Zhejiang, China

ARTICLE INFO

Keywords:
Video anomaly detection
Motion loss function
Variance attention strategy
Multi-layer reconstruction

ABSTRACT

In this paper, we propose a comprehensive framework for detecting anomalies in videos based on autoencoder (AE). Traditional AE models solely rely on input and final reconstruction, potentially limiting their capacity to fully utilize the intermediate neural network layers. To mitigate this limitation, we introduce a novel approach that concurrently trains the model using corresponding intermediate layers from both the encoder and decoder. This allows the model to capture more intricate features, thus enhancing its anomaly detection capabilities. Furthermore, we introduce a motion loss function that exclusively relies on original video frames rather than optical flow, rendering it more efficient and capable of extracting motion features. Additionally, we have devised a variance attention strategy that is parameter-free and can automatically directs our model's focus towards moving objects, further boosting the performance of our approach. Our experiments on three public datasets demonstrate the effectiveness and efficiency of our method in identifying abnormal events in complex scenarios. The code is publicly available at https://github.com/lsf2008/multRecLossAEPub.

1. Introduction

Video anomaly detection is a critical task in computer vision with diverse applications in surveillance and security. The identification of anomalous events in videos poses significant challenges due to the complexity involved in defining deviations.

The primary challenge in video anomaly detection lies in the scarcity of anomalous samples, leading to the common assumption that normal samples are abundant. Within this unsupervised framework, a prevalent method for anomaly detection based on reconstruction error is the autoencoder (AE) [1–5]. This one-class approach learns a compressed representation of normal data by evaluating the distinction between input and output utilizing neural networks. While effective at extracting essential video features, this method exclusively relies on the inputs and outputs of the neural network, neglecting other network layers, potentially resulting in inadequate information capture across different scales. To enhance anomaly detection, we propose simultaneous consideration of information from multiple layers and scales, enabling the model to capture more robust features and effectively gather information across various scales. Additionally, we introduce a motion reconstruction loss function aimed at capturing abnormal behavior in movement, facilitating the model's learning of distinctions between normal and abnormal video sequences. Furthermore, traditional AEs often assign equal weights to feature maps at any given location, which may not be suitable for anomaly detection. Instead, we design a method for automatically weighting features, free of parameters, enabling automatic weighting of the feature map according to its distribution. The main contributions are summarized as follows:

- We introduce a comprehensive framework for anomaly detection in videos using a multi-layer reconstruction AE, which concurrently considers information from multiple layers and scales of the neural network.
- We present a motion reconstruction loss function designed to capture abnormal movement behavior by relying on original video frames instead of optical flow, thus enhancing its efficiency.
- We propose a parameter-free variance attention strategy that directs the model to focus on the moving target so that the region where the moving target is located in the feature map has a larger weight.

The rest of this paper is organized as follows. Section 2 reviews the related works on anomaly detection. Section 3 describes our method.

* Corresponding author.
E-mail address: limax_2008@outlook.com (S. Li).

https://doi.org/10.1016/j.imavis.2024.105011
Received 5 October 2023; Received in revised form 3 February 2024; Accepted 29 March 2024
Available online 2 April 2024

Section 4 evaluates the proposed algorithm on the public datasets. Section 5 concludes the paper.

2. Related works

Anomaly detection can be broadly categorized into three types based on the quantity of anomalous samples employed: one-class-based methods, conventional AEs, and unsupervised methods.

One-class methods are trained with normal data to identify anomalies. These approaches often employ AEs for anomaly detection [6–9]. Initially, AEs are integrated with traditional methods. For instance, Hasan et al. [6] utilize handcrafted features as inputs to AEs for spatio-temporal consistency detection, while Tran and Hogg [7] use features extracted by AEs as inputs to one-class SVM for anomaly detection. In recent times, video anomaly detection is commonly approached as an end-to-end problem [10,11]. However, as observed by other researchers [4,12], AEs can also effectively reconstruct anomalies, leading to detection errors. To address this issue, several studies have endeavored to employ continuous memory representations to constrain the AE network [5,13,14]. The memory-based approach involves using a memory to record the patterns of normal samples. The detection process compares the differences between test samples and the normal samples in the memory, considering samples with significant differences as abnormal samples. These memory-based AEs can alleviate some drawbacks associated with AEs. However, the performance of memory-based methods may be limited by the memory size [15], potentially leading to restricted reconstruction with a small memory size. Additionally, the aforementioned methods primarily train the network using the encoder input and the decoder output, without integrating information from the intermediate layers in the AE network, leading to insufficient utilization of feature information across different network layers.

Another approach to anomaly detection involves the use of normal and anomalous data simultaneously. One such technique is Multiple Instance Learning (MIL), which identifies anomalous objects based on both normal and anomalous data. In the work by Sultani et al. [16], normal and anomalous videos are considered as bags and segments of videos as instances, respectively. To boost the detection accuracy, Zhu et al. [17] incorporate temporal context into the MIL ranking model using an attention block. Additionally, Zhang et al. [18] introduce a novel inner bag loss function that constrains the function space of weakly supervised problems. Tian et al. [19] propose Robust Temporal Feature Magnitude (RTFM) learning to enhance the robustness of MIL in various scenarios. However, due to sparse annotation information, it may be challenging to determine the instances that truly affect the class of the package. This issue could limit the model's performance and generalization.

The third category, unsupervised anomaly detection, is distinct in that it does not rely on any training data annotations. This presents a greater challenge compared to fully supervised, weakly supervised, or one-class supervised methods. Birant and Kut [20] propose a three-step method that utilizes a neighborhood-based mechanism to detect spatio-temporal anomalies. Wang, Meng, and Guo [21] employ three innovative hierarchical Bayesian models to learn low-level visual features, atomic activities, and interactions, enabling the resolution of numerous challenging visual surveillance tasks. Karadayi et al. [22] leverage a Long Short-Term Memory (LSTM) encoder and a Deep Neural Network (DNN)-based classifier to extract spatial and temporal contexts for anomaly detection. Zaheer et al. [8] introduce a generative cooperative learning method that comprises a generator and a discriminator to learn a frame-level anomaly score. Additionally, there are alternative approaches for video anomaly detection, which can be found in the literature of [23,24] for a more comprehensive overview.

3. Our work

Since anomalies often occur on moving objects, it is beneficial to enhance detection accuracy by eliminating the background [25,26]. However, this approach presents practical challenges and requires separate estimation of the video's background. To address this, we propose a variance attention strategy that dynamically weighs moving objects in the video, thereby obviating the need for background subtraction. Our method involves segmenting the videos into non-overlapping grid segments and training the model using a combined appearance and motion losses with our multi-layer reconstruction.

3.1. Problem statement

The framework of our method is depicted in Fig. 1. Let us consider a dataset containing n videos, denoted as $V = \{V_i\}_{i=1}^{n}$. Our approach is based on an AE architecture, comprising an encoder $\mathbf{E}$ and a decoder $\mathbf{D}$. Both the encoder and decoder consist of an equal number of layers, with the l-th layer of the encoder represented as f_l and decoder denoted as $f_l^{'}$. For a given training video V_i, we initially partition the video into non-overlapping segments. The continuous segments at the same location are then denoted as an input tensor $x = \{I_1, I_2, \cdots, I_{n_t}\}$, where I_i represents the i-th segmented frame, and n_t is the sequence length. Subsequently, we sequentially feed x into each layer $f_1, f_2, \cdots, f_{n_l}$ of our model $\mathbf{E}$, where n_l represents the number of encoder layers. This process yields the l-th layer feature map, denoted as $x_l = f_l(x_{l-1})$, with x_{l-1} being the output of the $(l-1)$-th layer. The final output $f_{n_l}(x_{n_l-1})$ of the encoder is then input into the layers $f_1^{'}, f_2^{'}, \cdots, f_{n_l}^{'}$ of the decoder $\mathbf{D}$ to obtain the reconstruction $x_i^{'} = f_i^{'}(x_{i-1}^{'})$ successively. Here, $x_i^{'}$ represents the corresponding reconstruction of x_i. Finally, the model is trained using multiple layer losses between x_i and the reconstruction $x_i^{'}$.

3.2. Video segmentation

It is useful to streamline the model learning process by reducing its complexity. To achieve this, we propose to segment the videos into small sequences. This approach enables our model to effectively learn and identify patterns of anomalies within the video data. Assuming a video V_i comprises T frames, each with a height H and width W, and a color channel C. The video can be represented as a tensor $V_i \in \mathbf{R}^{T\times H\times W\times C}$. We then divide this tensor into a sequence of non-overlapping tokens $x \in \mathbf{R}^{n_t\times h\times w\times C}$, where n_t denotes the sequence length, and h and w represent the height and width of each token, respectively.

3.3. Detection model

Our proposed model is founded on the AE network, specifically designed for processing video sequences. The architecture of the AE network is structured with fundamental downsampling and upsampling blocks. The downsampling block comprises three 3D convolution layers, each succeeded by batch normalization and activation layers. Additionally, the upsampling block integrates two 3D convolutional layers with a kernel size of 5 and one 3D convolutional layer with a kernel size of 3. In contrast, the downsampling block incorporates two 3D convolutional layers with a kernel size of 3 and one 3D convolutional layer with a kernel size of 1. The comprehensive architecture of the network is illustrated in Fig. 2.

3.4. Multi-layer reconstruction

Traditional AEs rely solely on input–output differences during training [4,5,12,13,15], limiting their capacity to capture intricate patterns in the data. This paper proposes an extension that employs multi-layer reconstruction to enhance anomaly detection accuracy. As illustrated in Fig. 3, the encoder comprises five layers, mirroring the structure of the decoder. The feature map of the l-th layer in the model is denoted as x_l. We introduce the input token x to the initial layer,

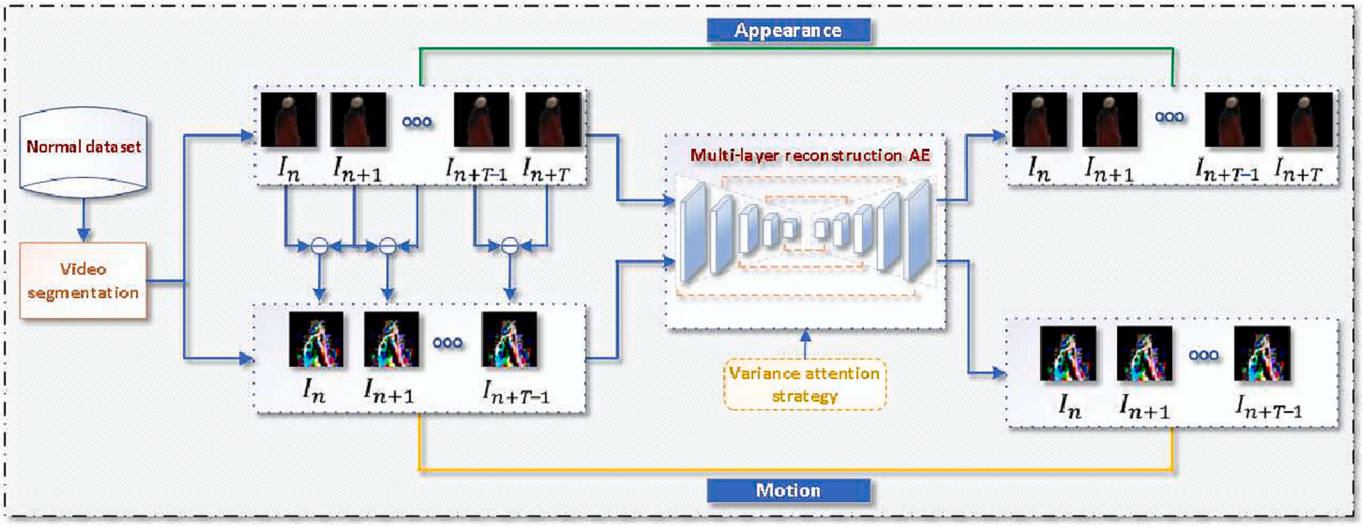

Fig. 1. The overall architecture of our method.

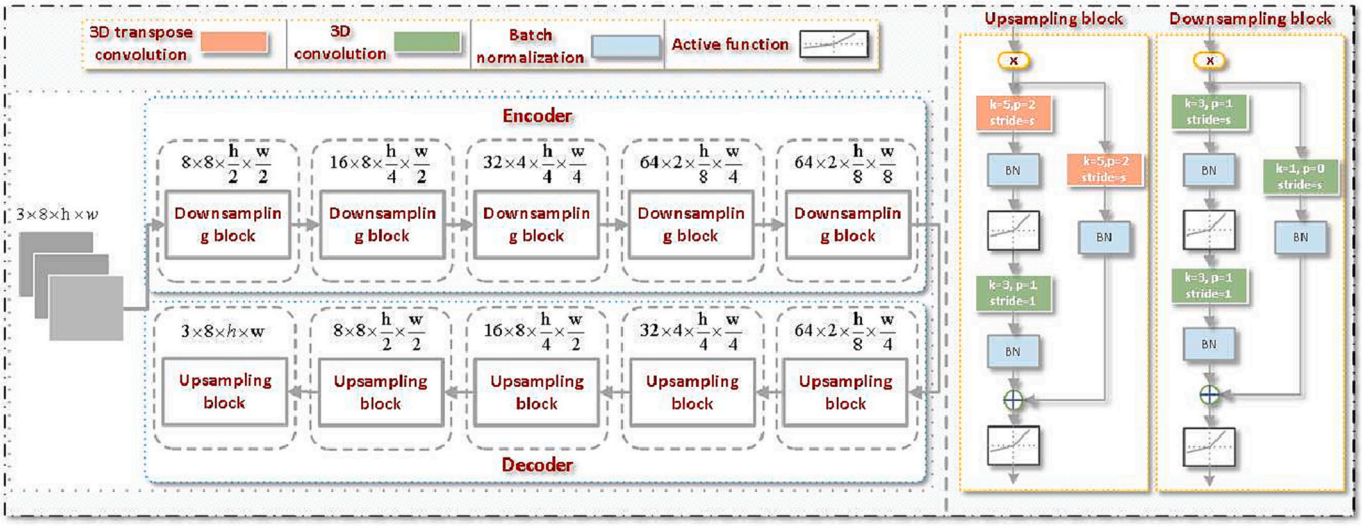

Fig. 2. Our network consists of upsampling and downsampling basic blocks. The left is the AE architecture and the right is the basic upsampling and downsampling blocks.

yielding a feature map x_1. Subsequently, x_1 is fed into the second layer, generating output x_2, and so forth for each encoder layer. The ultimate output of the last encoder layer is x_4, serving as input for the decoder. The decoder's first layer produces $x_4^{'}$, which is then upsampled by the second decoder layer to yield $x_3^{'}$, and so forth.

The loss function commonly utilized in traditional AE models involves the disparity between the input tensor x and its final reconstruction $x^{'}$. The formulation of this loss function can be expressed as follows:

$$\mathscr{L}_{\mathrm{sgl}} = \ell(x^{'}, x), \tag{1}$$

where $\ell(x^{'}, x)$ represents an arbitrary loss function measuring the difference between the reconstruction $x^{'}$ and the input tensor x. Denoting the encoder and decoder as $\mathbf{E}$ and $\mathbf{D}$ respectively, the reconstruction $x^{'}$ can be obtained as $x^{'} = \mathbf{D}(\mathbf{E}(x))$. The traditional AE loss function can then be written as:

$$\mathscr{L}_{\mathrm{sgl}} = \ell(\mathbf{D}(\mathbf{E}(x)), x). \tag{2}$$

As previously indicated, traditional AE models exclusively employ the input tensor x and output tensor $x^{'}$ for training. However, this methodology overlooks the intermediate layers within the model, thereby failing to comprehensively explore the features encapsulated within these layers. To address this limitation, we introduce a novel extension to the loss function, incorporating the information from these intermediate layers. Specifically, considering the output for the l-th layer f_l of the encoder $\mathbf{E}$ as $x_l = f_l(x_{l-1})$, and the corresponding reconstruction in the decoder $\mathbf{D}$ as $x_l^{'} = f_l^{'}(x_{l-1}^{'})$. The loss function for the l-th layer based on AE can then be expressed as:

$$\mathscr{L}_{sgl}^{l} = \ell\left(x_l^{'}, x_l\right) = \ell\left(f_l^{'}\left(x_{l-1}^{'}\right), f_l(x_{l-1})\right), \tag{3}$$

Moreover, we can generalize Eq. 3 to a specifically chosen subset of layers denoted by $\mathbf{S}$ in the formulation of the multi-layer loss function:

$$\mathscr{L}^{mul} = \frac{1}{|\mathbf{S}|}\sum_{l\in\mathbf{S}}\mathscr{L}_{sgl}^{l}. \tag{4}$$

Here, the set of layers $\mathbf{S}$ can encompass any combination of layers within the model, with $|\mathbf{S}|$ representing the cardinality of the set. The

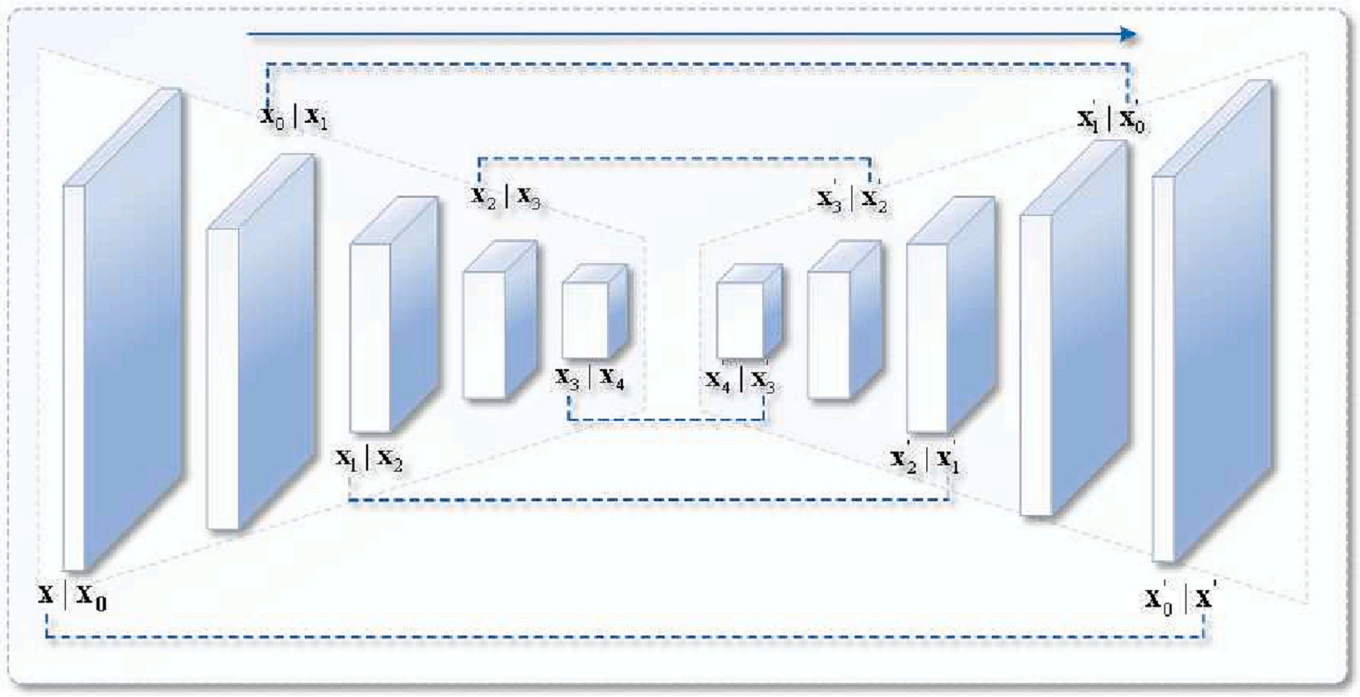

Fig. 3. Corresponding layers between encoder and decoder of our method.

extension of the loss function to encompass intermediate layers encourages the model to capture intricate patterns and representations across the entire network, thereby enhancing the performance.

3.5. Loss functions

In the process of training our model, we concurrently incorporate considerations for both appearance and motion losses.

3.5.1. Appearance loss

To compute the appearance loss function, we employ the RGB difference between the l-th tensor $x_l = \{I_1, I_2, \ldots, I_{n_t}\}$ and its corresponding reconstruction $x_l^{'} = \left\{I_1^{'}, I_2^{'}, \ldots, I_{n_t}^{'}\right\}$. Specifically, for the l-th layer of the model, the appearance loss function is defined as follows:

$$\mathscr{L}_{app}^{l} = \frac{1}{n_t \times C \times h \times w} \sum_{t=1}^{n_t} \left\| I_t^{'} - I_t \right\|_F^2, \tag{5}$$

Here, we assume that the token x is divided into n_t frames. The t-th frame is denoted as I_t, and its corresponding reconstructed frame is represented by $I_t^{'}$. The Frobenius norm, denoted by $\|\cdot\|_F$, is used for measuring distances. We extend the loss function to multiple layers in the model, as discussed earlier. The multi-layer appearance loss function can be formulated using Eq. 4.

$$\mathscr{L}_{app}^{mul} = \frac{1}{|\mathbf{S}_{app}|} \sum_{l \in \mathbf{S}_{app}} \mathscr{L}_{app}^{l}, \tag{6}$$

where $\mathbf{S}_{\text{app}}$ represents a combination of model layers. For instance, if the loss is computed using the first, second, and fourth layers, the set is denoted as $\mathbf{S}_{\text{app}} = [1, 2, 4]$. Specifically, we use [0] to refer to the input tensor of the model. $|\mathbf{S}_{\text{app}}|$ denotes the number of elements in the set $\mathbf{S}_{\text{app}}$.

3.5.2. Motion loss

Previous methods employ optical flow for motion features in anomaly detection [6,27]. However, it is computationally expensive. This paper introduces a simple motion loss function that calculates the gradient of adjacent frames along the time axis. This approach enhances efficiency and captures motion information between frames to improve detection performance.

Let $g_t = |I_t - I_{t-1}|$ represent the gradient map in the encoder between two adjacent frames $I_t \in x$ and $I_{t-1} \in x$, and $g_t^{'} = |I_t^{'} - I_{t-1}^{'}|$ denote the corresponding gradient map in the decoder between two adjacent frames $I_t^{'} \in x^{'}$ and $I_{t-1}^{'} \in x^{'}$. Given that a token x is comprised of n_t frames, there exist $n_t - 1$ gradient maps associated with this token. The motion loss function $\mathscr{L}_{\text{mot}}^{l}$ for the l-th layer can be defined as:

$$\mathscr{L}_{mot}^{l} = \frac{1}{(n_t - 1) \times C \times h \times w} \sum_{t=1}^{n_t - 1} \left\| g_t^{'} - g_t \right\|_F^2, \tag{7}$$

In addition to the motion loss function at a single layer, we extend its application to multiple layers within the model. This extension enables the capture of intricate motion patterns across diverse layers of the network.

$$\mathscr{L}_{mot}^{mul} = \frac{1}{|\mathbf{S}_{mot}|} \sum_{l \in \mathbf{S}_{mot}} \mathscr{L}_{mot}^{l}, \tag{8}$$

where $\mathbf{S}_{mot}$ refers to a selected set of model layers for which we want to compute the motion loss function.

3.5.3. Joint loss function

To enhance the overall performance of anomaly detection, we combine the appearance loss function with the motion loss function. The joint loss function is formally defined as follows:

$$\mathscr{L}_{join} = \alpha_1^{*} \mathscr{L}_{app}^{mul} + \alpha_2^{*} \mathscr{L}_{mot}^{mul}. \tag{9}$$

where α_1 and α_2 are the combination weights of appearance and motion loss function, respectively.

3.6. Variance attention strategy

It is widely acknowledged that anomalies frequently occur in regions where objects are in motion [28]. To mitigate the influence of irrelevant background and augment the significance of moving objects, a variance

attention strategy is utilized. This approach enables the model to concentrate on areas where motion undergoes significant changes, thereby enhancing its anomaly detection capability.

3.6.1. Variance attention along channel dimension

For descriptive purposes, the output feature of the l-th layer, denoted as f_l in the encoder **E**, is represented as $f_l(i_l, j_l, d_l, t_l)$ at the spatial coordinates (i_l, j_l) with a feature dimensionality of d_l and temporal dimensionality of t_l. The computation of the channel variance, $v_{(i_l,j_l,d_l)}$, at each spatial location (i_l, j_l) along the feature dimension is achieved using the following equation:

$$v^l_{(i_l,j_l,t_l)} = \frac{1}{C_l}\sum_{d_l=1}^{C_l}\left\| f(i_l,j_l,d_l,t_l) - \frac{1}{C_l}\sum_{d_l=1}^{C_l} f\left(i_l, j_l, d_l, t_l\right)\right\|_2, \tag{10}$$

where C_l represents the channel dimension of the l-th layer's output feature map. We then calculate the channel variance attention $w^l_{(i_l,j_l,t_l)}$ using softmax:

$$w^l_{(i_l,j_l,t_l)} = \frac{exp\left(v^l_{(i_l,j_l,t_l)}\right)}{\sum_{i=1,j=1}^{h_l,w_l} exp\left(v^l_{(i_l,j_l,t_l)}\right)}, \tag{11}$$

where h_l and w_l represent the height and width of the feature, respectively, with $i_l \in [1, h_l]$ and $j_l \in [1, w_l]$. This methodology enables the model to concentrate on regions characterized by motion changes, thereby enhancing its capability to detect anomalies.

3.6.2. Variance attention along time dimension

The previously mentioned channel variance attention is calculated along the feature dimension. Additionally, parallel operations along the time dimension are executed to acquire a temporal weight denoted as $w^l_{(i_l,j_l,d_l)}$. Assuming there are n_t frames within a single divided token, the first step involves computing the temporal variance:

$$v^l_{(i_l,j_l,d_l)} = \frac{1}{n_t}\sum_{t_l=1}^{n_t}\left\| f(i_l,j_l,d_l,t_l) - \frac{1}{n_t}\sum_{t_l=1}^{n_t} f\left(i_l, j_l, d_l, t_l\right)\right\|_2, \tag{12}$$

Then, we calculate the temporal variance attention by softmax:

$$w^l_{(i_l,j_l,d_l)} = \frac{exp\left(v^l_{(i_l,j_l,d_l)}\right)}{\sum_{i=1,j=1}^{h_l,w_l} exp\left(v^l_{(i_l,j_l,d_l)}\right)}. \tag{13}$$

Finally, we employ the channel variance attention $w^l_{(i_l,j_l,t_l)}$ to weigh each tensor $f_l(i_l, j_l, d_l, t_l)$ along the feature dimension. Simultaneously, temporal variance attention $w^l_{(i_l,j_l,d_l)}$ is applied to weigh the same tensor along the time dimension. The ultimate weighted feature map is the summation of the outcomes of both variance attention weighted feature maps. It is crucial to emphasize that the variance attention strategy can be extended to multiple layers. This strategy enables the model to concentrate on regions where motion undergoes massive changes along time and channels, thereby enhancing its capability to detect anomalies.

3.7. Anomaly detection score

The anomaly detection score comprises two components: one for appearance anomalies and another for motion anomalies. Both scores are directly derived from the loss function. Each anomaly score is normalized to a range between 0 and 1. If a sequence tensor $x = \{I_1, \cdots, I_{n_t}\}$ contains n_t frames, the ultimate anomaly score is computed from each normalized loss by:

$$S_x = \beta_1{}^* max_{i\in[0,\cdots,n_t]}\mathscr{L}_{app} + \beta_2{}^* max_{i\in[0,\cdots,n_t]}\mathscr{L}_{mot}. \tag{14}$$

Where β_i denotes the weight associated with each score, we set $\beta_1 = 0.5$ and $\beta_2 = 1$ for the Ped2 and ShanghaiTech datasets, and $\beta_1 = -0.1$ and $\beta_2 = 2$ for the Avenue dataset. The ultimate anomaly score for each clip is computed as the sum of the maximum value of each type loss across all tokens.

4. Experiments

We validate the effectiveness of our proposed method across three datasets: UCSD Ped2 [29], Avenue [30], and ShanghaiTech [31]. (a) The Ped2 dataset is the first to be proposed. It features objects moving parallel to the camera, with a resolution of 240 × 360. The dataset is organized into 16 clips for training and 12 clips for testing. The training dataset only comprises normally walking pedestrians, while the test dataset includes various individuals, such as non-pedestrians or those exhibiting abnormal behavior. (b) The Avenue dataset, originally introduced by Lu et al. [30], encompasses a total of 30,652 frames captured from an avenue on the CUHK campus. The dataset is partitioned into 16 videos for training and 21 videos for testing. (c) The ShanghaiTech dataset, the most challenging among the three, comprises 330 training videos and 107 testing videos, making it the most extensive dataset. It incorporates a total of 13 complex scenarios and over 130 anomalous events, including running, riding bicycles, fighting, etc.

4.1. Experiments setup

4.1.1. Evaluation criteria

The Area Under the Curve (AUC) serves as a predominant metric employed by researchers for assessing the performance of various anomaly detection methods [6,16,27,30,32–36]. AUC measures the performance of a method on a dataset, with a larger AUC score indicating a better performance of the test method. To further elucidate the effectiveness of our approach, we also use Equal Error Rate (EER) as an evaluation criterion. During the training phase, we utilize 16 consecutive frames, computing the reconstruction loss over this interval. For evaluating the effectiveness of our approach, we exclusively calculate the anomaly score for the ninth frame within the set of 16 testing frames, as employed in prior works [13,15].

4.1.2. Parameters setting and implementation details

The experiments are conducted on a PC server, and the code is implemented using PyTorch Lightning. In our implementation, we initially resize videos to 240×352 for the Ped2 and ShanghaiTech datasets and 300×528 for the Avenue dataset. Subsequently, each video sequence is partitioned into tokens of size $3 \times 8 \times h \times w$, where 3 denotes the channel, 8 represents the number of frames in a sequence, and h and w represent the token's height and width. To maintain uniformity across datasets, we fix the size to $3 \times 8 \times 48 \times 48$ for all videos. For training our model, we employ the AdamW optimizer with an initial learning rate of 0.0001, a decay rate of 0.0001, and a decay step of 80. The maximum batch size is set to 500, and training stopped if the validation AUC does not improve for 20 consecutive epochs. Unless explicitly specified, the parameters α_1 and α_2 in Eq. 9 are held constant at values of 1 and 25, respectively.

4.2. Qualitative results

We present detection samples obtained from Ped2, Avenue, and ShanghaiTech datasets. The detection process employs parameters identical to those used during training, as indicated earlier, with detection performed every 16 frames. The results are illustrated in Fig. 4, where the horizontal axis represents the video sequence index, and the vertical axis denotes the anomaly scores. The presence of cross-occurrences between normal and abnormal events introduces challenges in detecting anomalies, resulting in abrupt changes in the anomaly scores. This observation shows our method has the sensitivity to the

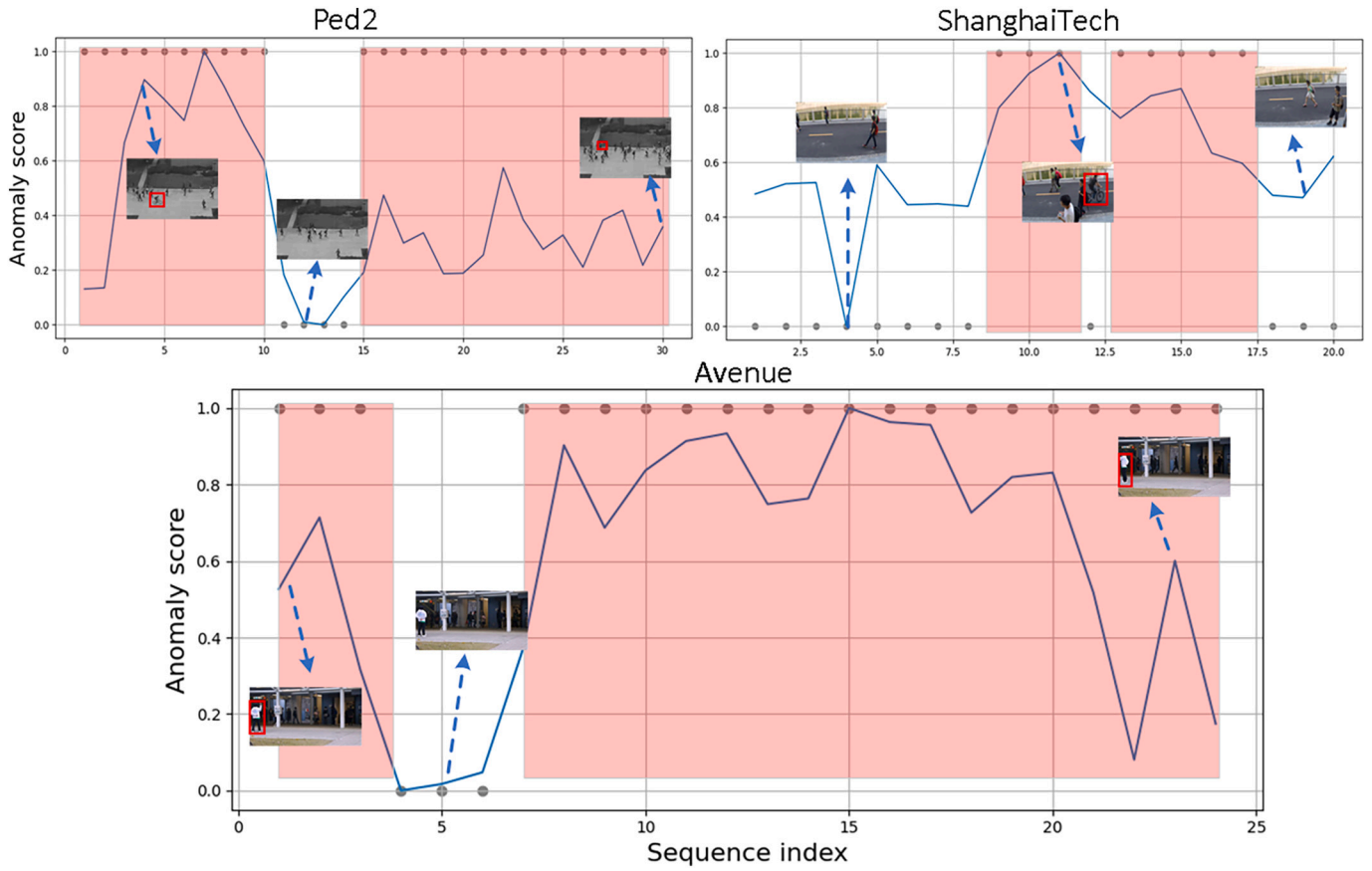

Fig. 4. Anomaly detection samples from Ped2, Avenue, and ShanghaiTech datasets.

changes in anomaly patterns and can detect anomalies successfully.

4.3. Quantitative evaluation

In this experiment, various anomaly detection methods are compared, including both traditional machine learning and deep learning approaches. Traditional machine learning methods include Scan Statistic (SS13) [37], Sparse Combination (SC13) [30], Spatio-Temporal Motion Context (STMC) [38], Combining Motion and Appearance (CMA16) [39], Winner-Take-All SVM (WTA-SVM) [7], MAP [40], MultiLevel Anomaly Detector (MLAD) [41], Temporally-coherent Sparse Coding (TSC) [42]. Additionally, deep learning methods are evaluated, such as Convolutional Auto Encoder (Conv-AE) [6], Combining Motion and Appearance (CMA16) [39], Winner-Take-All SVM (WTA-SVM) [7], Latent Space Autoregression (LSA) [43], Cluster AE [44], MemAE [13], Frame-Pred [31], Synthetic Temporal Anomaly (STA) [15], Residual Spatiotemporal Autoencoder (R-STAE) [11], Skip connected and Memory Guided Netowork (SMGNet) [14] and DR-cGAN [45].

The performance comparison of these methods on the UCSD Ped2 dataset is presented in the second column of Table 1. Our method demonstrates performance comparable to that of other methods within this dataset. Upon analysis, it is possible that the relative simplicity of the Ped2 dataset leads to overfitting in our method, given that our approach is trained based on divided segments.

The Avenue dataset is a greater challenge compared to the Ped2 datasets. This heightened difficulty is attributed to the presence of vibrations in specific testing videos and the existence of outliers in the training data, potentially complicating the learning process. Moreover, the infrequent occurrence of certain normal patterns in the training data poses challenges in their detection. Nevertheless, our method has

Table 1
Comparison with the state-of-the-art methods on Ped2, Avenue (Ave) and ShanghaiTech (ShT) datasets at frame level. Best performances in each dataset are highlighted as bold.

Methods	Ped2	Ave	ShT	PA	PS	AS	PAS
SS13 [37]	0.94	–	–	–	–	–	–
SC13 [30]	–	0.84	–	–	–	–	–
STMC [38]	0.86	–	–	–	–	–	–
Conv-AE [6]	0.90	0.70	–	0.80	–	–	–
CMA16 [39]	0.90	–	–	–	–	–	–
WTA-SVM [7]	0.97	–	–	–	–	–	–
Unmask [32]	0.82	0.81	–	0.815	–	–	–
ConvLSTM-AE [46]	0.881	0.77	–	0.825	–	–	–
TSC [42]	0.91	0.806	–	0.86	–	–	–
MAP [40]	0.92	0.87	–	0.895	–	–	–
Frame-Pred [31]	0.95	0.85	0.728	0.900	0.839	0.789	0.843
MLAD [41]	**0.983**	0.72	–	0.852	–	–	–
LSA [43]	0.954	–	0.725	–	0.840	–	–
MemAE [13]	0.941	0.833	0.712	0.887	0.827	0.773	0.829
CL [47]	0.978	0.864	0.716	0.921	0.847	0.790	0.853
Cluster AE [44]	0.965	0.86	0.733	0.913	0.849	0.797	0.853
MNAD [5]	0.902	0.828	0.698	0.865	0.800	0.763	0.809
SF [12]	0.965	0.847	0.759	0.906	0.862	0.803	0.857
STD [28]	0.967	0.871	0.737	0.919	0.852	0.804	0.858
R-STAE [11]	0.83	0.82	–	0.825	–	–	–
SMGNet [14]	0.86	0.84	–	0.85	–	–	–
DR-cGAN [45]	0.976	**0.91**	–	**0.943**	–	–	–
Our method	0.962	0.879	**0.778**	0.921	**0.87**	**0.829**	**0.873**

demonstrated remarkable performance on the Avenue dataset, as illustrated in Table 1, column three. Our approach achieved an AUC of 0.879, marking the second-best performance on this dataset. Method [45] exhibits a marginal superiority over our approach. However, it relies on optical flow, lacks end-to-end characteristics, and demands substantial computational resources for optical flow estimation.

The ShanghaiTech dataset is the most challenging dataset in this experiment, resulting in a low AUC value in the fourth column of Table 1. This can be attributed to the presence of intricate and diverse patterns that are difficult to detect, along with significant fluctuations in lighting conditions and camera angles. Despite these formidable challenges, our method exhibits exceptional performance, surpassing all other methods. This suggests the robustness and effectiveness of our proposed approach.

Although our method may exhibit suboptimal performance on the Ped2 and Avenue dataset, it has shown superior effectiveness on the ShanghaiTech datasets. To demonstrate the advantages of our proposed method, we combine the AUCs from various datasets and compute their mean values to represent the overall performance. Specifically, we denote the mean value of the Ped2 and Avenue combination as "PA," the mean value of the Ped2 and ShanghaiTech combination as "PS," the combination of Avenue and ShanghaiTech as "AS," and the mean value of all combinations as "PAS." The comparative results are detailed in the last four columns of Table 1. In three cases, our proposed method attains the highest AUC, thereby suggesting its effectiveness and robustness across a diverse combination of datasets.

Besides AUC, we also present a comparison utilizing the EER metric with existing works, as detailed in Table 2. On the Ped2 dataset, our method demonstrates comparability with state-of-the-art techniques, while excelling on the Avenue dataset. This observation implies the efficiency of our approach in achieving anomaly detection across diverse datasets.

4.4. Influence of parameters

To assess the influence of individual hyperparameters, we conduct experiments on the Ped2 and Avenue datasets, utilizing a resolution of 240×352. The model is trained for 500 epochs with an initial learning rate of 0.0001, which decreases by half every 80 steps until reaching the minimum rate of 0.00001. Throughout all experiments, the parameters in Eq. 9 remain constant at $\alpha_1 = 1$ and $\alpha_2 = 25$. Additionally, we employ the feature maps from the input tensor, layer 3, and layer 4 for both the encoder and decoder. This methodology allows for a systematic analysis of each hyperparameter's impact on the model's performance.

4.4.1. Token size

Our model is trainable with video tokens of varying sizes, ranging from smaller ones capturing specific details to larger ones encompassing higher-level features. We conduct an investigation on the Avenue dataset to analyze the impact of different token sizes, presenting the results on the left side of Fig. 5. It is crucial to note that smaller tokens may lead to overfitting due to limited information, while larger tokens may introduce more noise, making it challenging for the model to provide an accurate description. Our experiments demonstrate that optimal performance is achieved when using tokens of size $3 \times 16 \times 48 \times 48$, signifying a balance between capturing sufficient information and avoiding overfitting and noise interference.

Table 2
The comparison of EER with existing works on public datasets.

Dataset	Old [4]	MLAD [41]	STMC [38]	WTA-SVM [7]	Our
Ped2	0.07	0.05	0.25	0.11	0.10
Avenue	–	0.39	–	0.82	0.17
ShanghaiTech	–	–	–	–	0.31

4.4.2. Combination of loss functions

To optimize performance, we combine the motion and appearance loss functions, as depicted in Eq. 9. We explore the impact of different coefficients by testing various combinations, altering one coefficient at a time while maintaining the others at 1, using the Avenue dataset. Specifically, we set $\alpha_1 = 1$ and vary α_2 from 1 to 40. Additionally, we adopt a token size of $3 \times 8 \times 48 \times 48$ and utilize the input tensor, layers 3, and 4 for variance attention and multi-layer reconstruction. The results are presented on the right side of Fig. 5. It is evident that when $\alpha_2 \geq 5$, the performance of all models appears similar. However, for $\alpha_2 < 5$, the performance significantly diminishes. The optimal performance is achieved at $\alpha_2 = 25$. This observation indicates the significance of the motion loss function in video anomaly detection, with larger values of α_2 indicating a greater emphasis on the motion loss function.

4.5. Combination of scores

We compute the anomaly score for each clip by summing the maximum value of each type of loss across all tokens, as shown in Eq. 14. To explore the impact of different weights, we assess various combinations by adjusting one weight at a time while keeping the other constant on the Ped2 dataset. The parameters utilized in this experiment are consistent with those mentioned earlier. The comparative results are presented in Table 3. In this experiment, we denote (β_1, β_2) as the weights for appearance and motion, respectively. We observe that the performance improves when β_2 exceeds β_1, suggesting that motion score holds greater significance than appearance score.

4.6. Ablation investigation

In this section, we conduct ablation studies to investigate the impact of various components in our proposed model. All experiments utilize the Ped2 dataset with identical parameter settings as previously described. Our model consists of four key components: appearance loss, motion loss, multi-layer reconstruction, and a variance attention strategy. We maintain a fixed token size of $3 \times 8 \times 48 \times 48$, apply weight to the feature maps of layers [0, 3, 4], and compute reconstructions at these specific layers. The ablation studies, as outlined in Table 4, emphasize the critical role of both loss functions in achieving optimal performance. Our experiments demonstrate that the removal of either loss function leads to a significant reduction in performance, highlighting the indispensability of both components. Moreover, our findings underscore the significance of the variance attention module, revealing it as a key component in our model. The removal of this module results in a notable drop in performance.

4.6.1. Multi-layer reconstruction

In our proposed model, we compute the loss function across multiple layers, as opposed to relying solely on the input tensor and output reconstruction. This approach enhances the model's flexibility by allowing the combination of various layers to calculate the loss functions. For example, the loss function is computed using the first, fourth, and last layers, denoted as $\mathbf{S} = [1, 4, 5]$. To assess the significance of multi-layer reconstruction, we randomly select 10 layer combinations and run our model using these selections. Additionally, we implement the variance attention strategy on both the input tensor and the feature maps of the third and fourth layers.

The experiments are conducted on the Ped2 dataset, and the results are detailed in Table 5. As previously noted, the baseline case [0] corresponds to the original AE, utilizing only the input tensor and the final output reconstruction. Across most scenarios, the performance of the model with multi-layer reconstruction surpasses that of the original AE. This observation suggests that optimal performance necessitates training the model with multiple layers of reconstruction. Notably, we observe suboptimal performance when the combination set excluded the original input tensor. This deficiency arises because the anomaly score is

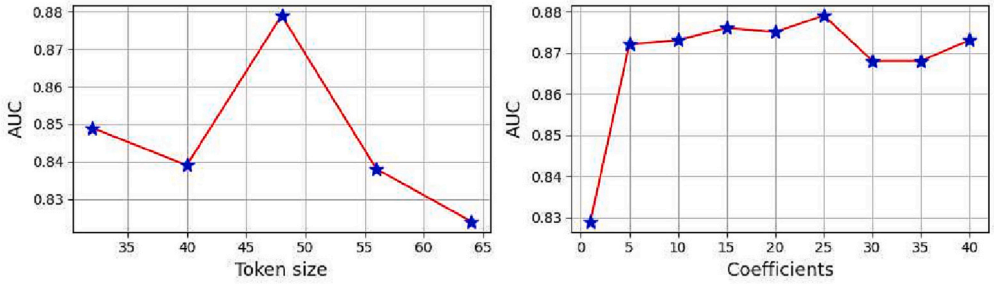

Fig. 5. Comparisons with various parameter settings: On the left, we evaluate performance across different token sizes, while on the right, we assess the impact of varying α_2 values while maintaining α_1 fixed at 1. The vertical axes represent the AUC, and the horizontal axis on the left corresponds to token size, while on the right, it represents the coefficients of α_2.

Table 3
Comparison of AUCs with different anomaly score weights.

weights	(1,1)	(1,2)	(1,3)	(1,5)	(2,5)
AUC	0.888	0.962	0.950	0.944	0.950
weights	(5,1)	(5,2)	(5,3)	(10,1)	(4,1)
AUC	0.618	0.690	0.770	0.558	0.638

Table 4
Ablation study on the Ped2 dataset.

App. Loss	Mot. Loss	Multi. Rec.	Var. Att.	AUC on Ped2
✓				0.864
	✓			0.837
✓	✓			0.923
✓	✓	✓		0.955
✓	✓	✓	✓	0.962

Table 5
Comparison of AUC with different reconstruction layers.

Layers	[0]	[0, 1]	[0, 1, 2]	[0, 3, 4]	[0, 2, 3]
AUC	0.923	0.929	0.940	0.962	0.940
Layers	[0, 1, 4]	[3,4]	[2–4]	[1,4]	[0,1,2,3,4]
AUC	0.939	0.903	0.919	0.919	0.929

computed solely based on the difference between the input tensor and the final output reconstruction, which neglects the utilization of intermediate layers.

4.6.2. *Variance attention strategy*

In our model, we integrate a variance attention strategy to intensify focus on moving objects. This is accomplished by assigning weights to the feature maps across various layers within the embedded space. To assess the effectiveness of the variance attention strategy, we conduct experiments comparing the model's performance with and without its incorporation. The results are detailed in Table 6. In most cases, the model featuring the devised variance attention strategy surpasses its counterpart lacking this strategy. This observation implies that the proposed variance attention strategy is effective in enhancing the anomaly detection performance.

Table 6
Comparison between the method using variance attention strategy and the one that does not use it.

α_2	0.1	0.5	1	5	10	15
AUC w/o weight	0.918	0.923	0.924	0.955	0.913	0.925
AUC w/ weight	0.947	0.930	0.962	0.948	0.930	0.944
α_2	20	25	30	35	40	45
AUC w/o weight	0.925	0.925	0.924	0.923	0.928	0.929
AUC w/ weight	0.959	0.962	0.947	0.917	0.953	0.941

Furthermore, to demonstrate the effectiveness of our variance attention strategy, we conduct experiments on a randomly selected testing video from the Ped2 dataset, computing variance maps at the first layer of the encoder. In this experimental setup, we feed the encoder with a tensor $x \in \mathbb{R}^{3\times8\times48\times48}$ and compute both channel and temporal variance maps at the first layer. The dimensions of the channel and temporal maps are $8 \times 24 \times 24$ (For simplicity, the batch dimension is omitted in all tensor representations above). The 3rd and 4th maps are visualized in Fig. 6. The visualization indicates that our variance attention map can accurately identify moving objects, allocating more attention to them, thereby improving motion detection accuracy.

4.7. *Efficiency of our model*

In this section, we present a comparison of the running time performance of our method with other publicly available approaches on the Ped2 dataset. It is crucial to acknowledge that running time can be influenced by diverse hardware configurations and video resolutions. Our method is deployed on a PC server equipped with an NVIDIA GeForce Titan 3080Ti card and an Intel(R) Core(TM) i7-11700K @ 3.60GHz processor. The video frames are resized to dimensions of 240 × 352 for our method. The Ped2 dataset comprises 12 test videos totaling 2010 frames. To ensure precision, we execute our method 10 times on the Ped2 dataset and compute the average time consumption as the final running time. Table 7 lists the running time comparison on the UCSD Ped2 dataset.

The computational time of our method across all test videos approximates 13.05 s, indicating a per-frame running time of 0.0065 s. Notably, methods such as [48,49], which also rely on tokens and employ deep learning architectures for anomaly detection, exhibit longer running time compared to our approach. Specifically, our method's running time is approximately 1153.8 times faster than [48] and 5.5 times faster than [49].

5. Conclusion

This paper introduces a framework designed for the detection of anomalies in videos, employing an AE exclusively trained on normal data. Given that anomalies typically manifest on moving objects, we have devised a variance attention strategy to accentuate the weighting assigned to such objects. To fully leverage the capabilities of the intermediate layers within the AE model, we advocate the simultaneous training of the model using multiple corresponding layers from both the encoder and decoder. This approach facilitates the exploration of a diverse range of features and representations present across multiple layers. Additionally, we have proposed a straightforward loss function capable of accurately capturing object motion features based on the gradient between adjacent frames. This function proves highly efficient for motion detection and holds applicability to various video-based

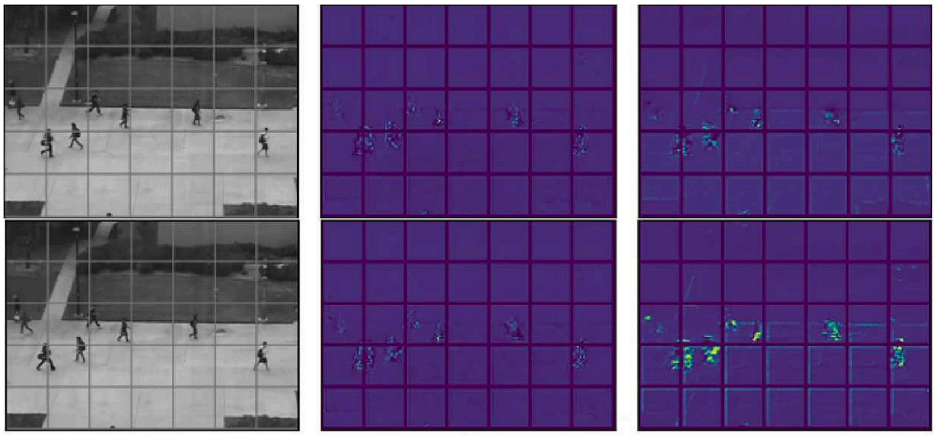

Fig. 6. Visualization of variance attention strategy (The brighter the color, the greater the weight). (a) Original input video frame. (b) Channel variance attention map. (c) Temporal variance attention map.

Table 7
Comparison of running time on the Ped2 dataset (seconds per frame).

Methods	CPU	GPU	Time
Unmask [32]	N/A	N/A	0.4
AMDN [48]	2.1GHz	Nvidia Auadro K4000	7.5
ALOCC [2]	2.1GHz	Nvidia GTX TITAN Xp	0.033
Deep-anomaly [50]	2.1GHz	Nvidia TITANX	0.036
MemAE [13]	N/A	Nvidia GeForce 1080Ti	0.0262
NM-GAN [49]	2.1GHz	Nvidia TITANX	0.036
Our method	3.6GHz	Nvidia Titan 3080Ti	0.0065

scenarios. The AUC metrics on public datasets, including Ped2, Avenue, and ShanghaiTech, yield scores of 0.962, 0.879, and 0.778, respectively. Notably, our method surpasses the majority of existing approaches in terms of both accuracy and efficiency.

CRediT authorship contribution statement

Shifeng Li: Writing – original draft, Writing – review & editing. **Yan Cheng:** Data curation. **Liang Zhang:** Validation. **Xi Luo:** Formal analysis. **Ruixuan Zhang:** Validation.

Declaration of competing interest

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

Data availability statement

The datasets used in our study include UCSDPed2 dataset [29], Avenue dataset [30], and ShanghaiTech dataset [31]. These datasets are obtained from publicly resources: Ped2 dataset (http://www.svcl.ucsd.edu/projects/anomaly/dataset.html), Avenue dataset (https://www.cse.cuhk.edu.hk/leojia/projects/detectabnormal/dataset.html), ShanghaiTech dataset (https://svip-lab.github.io/dataset/campus_dataset.html). The code of our method is available at https://github.com/lsf2008/multRecLossAEPub.

Acknowledgment

This work was jointly supported by the National Natural Science Foundation of China (61402049), Science and Technology Research Project of the Department of Education of Liaoning Province (LJKZ1019) and Social Science Planning Fund of Liaoning Province (L21BGL002).

References

[1] S.M. Erfani, S. Rajasegarar, S. Karunasekera, C. Leckie, High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recogn. 58 (2016) 121–134.
[2] M. Sabokrou, M. Khalooei, M. Fathy, E. Adeli, Adversarially learned one-class classifier for novelty detection, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3379–3388.
[3] L. Ruff, N. Görnitz, L. Deecke, S.A. Siddiqui, R.A. Vandermeulen, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proceedings of the 35th International Conference on Machine Learning Vol. 80, 2018, pp. 4390–4399.
[4] M.Z. Zaheer, J. Lee, M. Astrid, S. Lee, Old is gold: redefining the adversarially learned one-class classifier training paradigm, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14171–14181.
[5] H. Park, J. Noh, B. Ham, Learning memory-guided normality for anomaly detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14360–14369.
[6] M. Hasan, J. Choi, J. Neumann, A.K. Roy-Chowdhury, L.S. Davis, Learning temporal regularity in video sequences, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 733–742.
[7] H. Tran, D.C. Hogg, Anomaly detection using a convolutional winner-take-all autoencoder, in: British Machine Vision Conference, 2017.
[8] M. Z. Zaheer, A. Mahmood, M. H. Khan, M. Segù, F. Yu, S. Lee, Generative cooperative learning for unsupervised video anomaly detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14724–14734.
[9] Z. Yang, J. Liu, Z. Wu, P. Wu, X. Liu, Video event restoration based on keyframes for video anomaly detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, IEEE, Vancouver, BC, Canada, June 17–24, 2023, pp. 14592–14601.
[10] B. Zong, Q. Song, M.R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: 6th International Conference on Learning Representations, 2018.
[11] K. Deepak, S. Chandrakala, C.K. Mohan, Residual spatiotemporal autoencoder for unsupervised video anomaly detection, Signal Image Video Process. 15 (1) (2021) 215–222.
[12] M. Astrid, M.Z. Zaheer, J. Lee, S. Lee, Learning not to reconstruct anomalies, in: 32nd British Machine Vision Conference, 2021, p. 279.
[13] D. Gong, L. Liu, V. Le, B. Saha, M.R. Mansour, S. Venkatesh, A. van den Hengel, Memorizing normality to detect anomaly: memory-augmented deep autoencoder for unsupervised anomaly detection, in: 2019 IEEE/CVF International Conference on Computer Vision, 2019, pp. 1705–1714.
[14] S. Chandrakala, V. Srinivas, K. Deepak, Residual spatiotemporal autoencoder with skip connected and memory guided network for detecting video anomalies, Neural. Process. Lett. 53 (6) (2021) 4677–4692.
[15] M. Astrid, M.Z. Zaheer, S. Lee, Synthetic temporal anomaly guided end-to-end video anomaly detection, in: IEEE/CVF International Conference on Computer Vision Workshops, 2021, pp. 207–214.
[16] W. Sultani, C. Chen, M. Shah, Real-world anomaly detection in surveillance videos, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition Society, 2018, pp. 6479–6488.
[17] Y. Zhu, S.D. Newsam, Motion-aware feature for improved video anomaly detection, in: 30th British Machine Vision Conference, 2019, p. 270.
[18] J. Zhang, L. Qing, J. Miao, Temporal convolutional network with complementary inner bag loss for weakly supervised anomaly detection, in: In: 2019 IEEE International Conference on Image Processing, IEEE, 2019, pp. 4030–4034.

[19] Y. Tian, G. Pang, Y. Chen, R. Singh, J. W. Verjans, G. Carneiro, Weakly-supervised video anomaly detection with robust temporal feature magnitude learning, in: 2021 IEEE/CVF International Conference on Computer Vision, pp. 4955–4966.
[20] D. Birant, A. Kut, Spatio-temporal outlier detection in large databases, J. Comput. Inf. Technol. 14 (4) (2006) 291–297.
[21] X. Wang, X. Ma, W.E.L. Grimson, Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models, IEEE Trans. Pattern Anal. Mach. Intell. 31 (3) (2009) 539–555.
[22] Y. Karadayi, Unsupervised anomaly detection in multivariate spatio-temporal datasets using deep learning, in: V. Lemaire, S. Malinowski, A. J. Bagnall, A. Bondu, T. Guyet, R. Tavenard (Eds.), Advanced Analytics and Learning on Temporal Data 2019, Vol. 11986, 167–182.
[23] A.B. Mabrouk, E. Zagrouba, Abnormal behavior recognition for intelligent video surveillance systems: a review, Expert Syst. Appl. 91 (2018) 480–491.
[24] R. Nayak, U.C. Pati, S.K. Das, A comprehensive review on deep learning-based methods for video anomaly detection, Image Vis. Comput. 106 (2021) 104078.
[25] S. Li, Y. Cheng, Y. Tian, Y. Liu, Anomaly detection based on superpixels in videos, Neural Comput. & Applic. 34 (15) (2022) 12617–12631.
[26] S. Li, Y. Cheng, Y. Liu, Y. Yang, Fast anomaly detection based on 3d integral images, Neural. Process. Lett. 54 (2) (2022) 1465–1479.
[27] J.T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, R.S.M. Goh, Anomalynet: an anomaly detection network for video surveillance, IEEE Trans. Inf. Forensics Secur. 14 (10) (2019) 2537–2550.
[28] Y. Chang, Z. Tu, W. Xie, B. Luo, S. Zhang, H. Sui, J. Yuan, Video anomaly detection with spatio-temporal dissociation, Pattern Recogn. 122 (2022) 108213.
[29] V. Mahadevan, W. Li, V. Bhalodia, N. Vasconcelos, Anomaly detection in crowded scenes, in: The Twenty-Third IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1975–1981.
[30] C. Lu, J. Shi, J. Jia, Abnormal event detection at 150 FPS in MATLAB, in: IEEE International Conference on Computer Vision, 2013, pp. 2720–2727.
[31] W. Liu, W. Luo, D. Lian, S. Gao, Future frame prediction for anomaly detection - a new baseline, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6536–6545.
[32] R.T. Ionescu, S. Smeureanu, B. Alexe, M. Popescu, Unmasking the abnormal events in video, in: IEEE International Conference on Computer Vision, 2017, pp. 2914–2922.
[33] W. Liu, W. Luo, Z. Li, P. Zhao, S. Gao, Margin learning embedded prediction for video anomaly detection with A few anomalies, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 3023–3030.
[34] R.T. Ionescu, F.S. Khan, M. Georgescu, L. Shao, Object-centric auto-encoders and dummy anomalies for abnormal event detection in video, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7842–7851.
[35] T. Nguyen, J. Meunier, Anomaly detection in video sequence with appearance-motion correspondence, in: 2019 IEEE/CVF International Conference on Computer Vision, 2019, pp. 1273–1283.
[36] G. Pang, C. Shen, A. van den Hengel, Deep anomaly detection with deviation networks, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 353–362.
[37] Y. Hu, Y. Zhang, L.S. Davis, Unsupervised abnormal crowd activity detection using semiparametric scan statistic, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 767–774.
[38] Y. Cong, J. Yuan, Y. Tang, Video anomaly search in crowded scenes via spatio-temporal motion context, IEEE Trans. Inf. Forensics Secur. 8 (10) (2013) 1590–1599.
[39] Y. Zhang, H. Lu, L. Zhang, X. Ruan, Combining motion and appearance cues for anomaly detection, Pattern Recogn. 51 (2016) 443–452.
[40] S. Li, C. Liu, Y. Yang, Anomaly detection based on maximum a posteriori, Pattern Recogn. Lett. 107 (2018) 91–97.
[41] H. Vu, T.D. Nguyen, T. Le, W. Luo, D.Q. Phung, Robust anomaly detection in videos using multilevel representations, in: The Thirty-Third AAAI Conference on Artificial Intelligence, 2019, pp. 5216–5223.
[42] W. Luo, W. Liu, S. Gao, A revisit of sparse coding based anomaly detection in stacked RNN framework, in: IEEE International Conference on Computer Vision, 2017, pp. 341–349.
[43] D. Abati, A. Porrello, S. Calderara, R. Cucchiara, Latent space autoregression for novelty detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 481–490.
[44] Y. Chang, Z. Tu, W. Xie, J. Yuan, Clustering driven deep autoencoder for video anomaly detection, in: ECCV Vol. 12360, 2020, pp. 329–345.
[45] T. Ganokratanaa, S. Aramvith, N. Sebe, Video anomaly detection using deep residual-spatiotemporal translation network, Pattern Recogn. Lett. 155 (2022) 143–150.
[46] W. Luo, W. Liu, S. Gao, Remembering history with convolutional LSTM for anomaly detection, in: 2017 IEEE International Conference on Multimedia and Expo, 2017, pp. 439–444.
[47] K. Doshi, Y. Yilmaz, Continual learning for anomaly detection in surveillance videos, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1025–1034.
[48] D. Xu, Y. Yan, E. Ricci, N. Sebe, Detecting anomalous events in videos by learning deep representations of appearance and motion, Comput. Vis. Image Underst. 156 (2017) 117–127.
[49] D. Chen, L. Yue, X. Chang, M. Xu, T. Jia, NM-GAN: noise-modulated generative adversarial network for video anomaly detection, Pattern Recogn. 116 (2021) 107969.
[50] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, R. Klette, Deep-anomaly: fully convolutional neural network for fast anomaly detection in crowded scenes, Comput. Vis. Image Underst. 172 (2018) 88–97.