sensors

MDPI

Article

Video Anomaly Detection Based on Convolutional Recurrent AutoEncoder

Bokun Wang [1] and Caiqian Yang [2,*]

[1] College of Civil Engineering and Mechanics, Xiangtan University, Xiangtan 411100, China; 201921002175@smail.xtu.edu.cn
[2] School of Civil Engineering, Southeast University, Nanjing 210096, China
* Correspondence: ycqjxx@seu.edu.cn

Abstract: As an essential task in computer vision, video anomaly detection technology is used in video surveillance, scene understanding, road traffic analysis and other fields. However, the definition of anomaly, scene change and complex background present great challenges for video anomaly detection tasks. The insight that motivates this study is that the reconstruction error for normal samples would be lower since they are closer to the training data, while the anomalies could not be reconstructed well. In this paper, we proposed a Convolutional Recurrent AutoEncoder (CR-AE), which combines an attention-based Convolutional Long–Short-Term Memory (ConvLSTM) network and a Convolutional AutoEncoder. The ConvLSTM network and the Convolutional AutoEncoder could capture the irregularity of the temporal pattern and spatial irregularity, respectively. The attention mechanism was used to obtain the current output characteristics from the hidden state of each Covn-LSTM layer. Then, a convolutional decoder was utilized to reconstruct the input video clip and the testing video clip with higher reconstruction error, which were further judged to be anomalies. The proposed method was tested on two popular benchmarks (UCSD ped2 Dataset and Avenue Dataset), and the experimental results demonstrated that CR-AE achieved 95.6% and 73.1% frame-level AUC on two public datasets, respectively.

Keywords: video anomaly detection; deep learning; convolutional long–short-term memory; convolutional autoencoder

Citation: Wang, B.; Yang, C. Video Anomaly Detection Based on Convolutional Recurrent AutoEncoder. *Sensors* **2022**, *22*, 4647. https://doi.org/10.3390/s22124647

Academic Editor: Krzysztof Holak

Received: 12 May 2022
Accepted: 14 June 2022
Published: 20 June 2022

1. Introduction

In order to improve the safety of people's lives and public property, video surveillance systems have been widely installed in public places such as train stations, airports, hospitals, markets, schools, and resident centers. The main goal of social public safety risk prevention and control is to detect abnormal events accurately and timely. However, it is a tedious process to monitor the surveillance videos at a continuously faster rate, which leads to inefficient utilization of surveillance cameras and requires human presence for monitoring. Hence, video anomaly detection has recently become an important research problem in computer vision [1,2]. Given a surveillance video clip, the aim of frame-level video anomaly detection is to identify frames where there is an event or behavior that differs from the expectations or that appears infrequent. These abnormal events usually include fights, riots, violations of traffic rules, sdtrampling, holding arms, and abandoning luggage. However, video anomaly detection in general is a vast, crucial, and challenging research topic due to the ambiguity of anomaly definitions, the paucity of anomalous data, and the complex environmental background.

In general, current research work of video anomaly detection contains two procedures: feature extraction and model learning [3]. Feature extraction can be achieved by handcrafteded feature technology or automatic feature extraction technology (features-based representation learning or deep learning). For the model learning procedure, normal samples are used for learning the detection model, and then, the testing samples that do

not conform to the learned model are judged as abnormal events. There are three main categories of feature extraction approaches. (1) Trajectory-based methods [4]: Various methods track the target to obtain trajectory features and achieve satisfactory detection results for anomalies in both speed and direction, but target tracking in dense scenes is a big problem. For example, the authors in [5] studied the detection of abnormal vehicle trajectories, such as illegal U-turns. The authors in [6] extracted human skeleton trajectory patterns and were thus limited to detecting human behavioral anomalies. (2) Methods based on variable features [7,8]: Various methods take video frames as a whole and extract some simultaneous or mid-level features such as spatiotemporal gradients, histogram of gradient, optical flow, etc., which are effective in moderately crowded and dense environments. In [9], the authors proposed to associate the optical flows between multiple frames to capture short-term trajectories and to introduce the histogram-based shape descriptor to describe such short-term trajectories. (3) Grid feature-based method [10]: For the reason that each grid can be evaluated separately, this method often divides the video frame into multiple small grids through dense sampling, and then extracts overlapping features from the subdivided grids. As an example, Roshtkhari employed a probability density function to encode spatio-temporal configurations of video volumes based on spatio-temporal gradient features.

Furthermore, it could be divided into three categories by different model learning strategies. (1) Cluster-based methods [11]: these methods are often based on the hypotheses that normal samples belong to a category or are closer to one cluster center, while the abnormal samples do not belong to any category or away from any cluster center, and then the normal samples are clustered to build the detection model. In [12], the set of features generated by a convolutional autoencoder are clustered, and a one-versus-rest classifier is trained that discriminates between the clusters to detect the anomaly. (2) Sparse reconstruction based method [13]: This type of method assumes that the sparse linear combination of normal patterns can represent normal activities with the smallest reconstruction error, and because there is no abnormal activity in the training dataset, it can represent abnormal patterns with larger reconstruction errors. One such method is introduced by Hasan [14], where the use of combining 2D convolutions to autoencoders was produced, wherein the 2D convolutions were taken as input specific raw video segments. (3) The method based on the probability model [15]: This method considers that normal samples that conform to a certain probability distribution, while abnormal samples do not match this distribution. In [15], the detection of anomalies in a video is based on the hypothesis that the normal samples can be associated with at least one Gaussian component of a Gaussian Mixture Model (GMM), while anomalies do not belong to any Gaussian component.

Recently, the latest progress of deep learning has proven the obvious advantages of artificial intelligence-based methods and not be confined to many computer vision applications [16]. As one of the topic tasks in computer vision, video anomaly detection is no exception. Unlike the hand-crafteded feature-based methods, these deep learning-based methods often use pre-trained neural network architecture to extract high-level features, or build an end-to-end anomaly detection model with existing network architecture. For the latter idea [1,11–23], the feature extraction step and model building step are jointly optimized with one network. These end-to-end deep neural networks contain Deep Auto-Encoders (AE, Auto-Encoder) [14], Deep Siamese Networks (DSN) [17], and Generative Adversarial Nets (GAN) [18]. However, these network models are often designed for other tasks such as generative models, compression, etc., rather than for anomaly detection tasks.

Different from the possible solutions discussed earlier, in this paper, we propose a new deep learning-based method called the Convolutional Recurrent AutoEncoder (CR-AE) for video anomaly detection. Specifically, the proposed method is based on the combination of an attention-based Convolutional Long–Short-Term Memory (ConvLSTM) network and the convolutional encoder of the AutoEncoder, which are employed to capture the irregularity of the temporal pattern and spatial irregularity, respectively. Then, the convolutional decoder of the AutoEncoder is utilized to remodel the input video clip, and the reconstruction

errors are further employed to detect abnormal frames. This is due to the reason that if the CR-AE has never observed a similar abnormal pattern before, it may not be able to reconstruct the input video clip well. Before our work, Hasan et al. [14] proposed to learn temporal regular patterns using a convolutional AutoEncoder with limited supervision to detect the video abnormal temporal events. Different from Hasan's work, the proposed method could simultaneously detect the spatial and temporal anomalies. In addition, frame-level annotation is carried out on two public datasets called the UCSD ped2 dataset and the ShanghaiTech dataset to evaluate anomaly detection performance of our method. The experimental results demonstrate that our method has good characteristics of strong generalization ability and outperforms the state-of-the-art methods.

In summary, the main contributions of this study are as follows:

1. We proposed an end-to-end deep learning framework for anomaly detection called Convolutional Recurrent AutoEncoder (CR-AE) for video anomaly detection. It is established by encoding the spatial regularity and temporal pattern with two common network architectures. They are the attention-based Convolutional Long–Short-Term Memory (ConvLSTM) network and the convolutional AutoEncoder (ConvAE). To the extent of our knowledge, this is the first time that the hybrid architectures of the attention-based ConvLSTM and ConvAE have been considered for video anomaly detection.
2. We adopted only a network to simultaneously detect the spatial and temporal anomaly to replace the conventional two-stream network. Compared with the conventional two-stream network, the CR-AE need not extract optical flow and train the weights of the two architectures.
3. We extensively evaluated our approach on the publicly available video anomaly detection datasets. The experiment demonstrates that our approach attains superior results compared to the state-of-the-art methods.

The remainder of this paper is structured as follows: Section 2 summarizes the related literature about existing anomaly detection. Section 3 describes the architecture of the proposed approach. Experimental evaluation on two public experiments is given in Section 4. Finally, Section 5 draws the conclusions of this paper.

2. Related Work

This section outlines the previous works on existing video anomaly detection methods, which include the hand-crafted feature-based and deep learning-based anomaly detection method.

2.1. Hand-Crafted Feature-Based Anomaly Detection Method

Early research on video anomaly detection adopted the hand-crafted features to represent the appearance and movement characteristics of pedestrians, and then machine learning method was used to learn the anomaly detection model. According to whether the object detection and object tracking procedure is adopted, these methods fall into two broad categories: anomaly detection methods based on trajectories and anomaly detection methods on cuboids.

Each trajectory represents the movement of a target as a sequence of image coordinates. The main idea of trajectory-based methods is based on the assumption that the anomalous trajectories of the abnormal events differ from the normal patterns. Junejo et al. [24] utilized the size, position, and speed of the trajectory as the feature to represent the event and to train a dynamic Bayesian network for abnormal behavior detection. Similarly, Kang et al. [25] proposed to utilize trajectory features to build a hidden Markov model to achieve an anomaly detection model. Similarly, Wang et al. [26] projected a dense trajectory algorithm that first densely sampled the feature points, then extracted the point trajectory features and encoded them, and then classified them through support vector machines in the end. After that, Wang improved the feature regularization and encoding method, and employed an improved method dense trajectory algorithm to represent the event in the video [27]. However, the detection result of the trajectory-based method depends on the

accuracy of the object tracking method. Furthermore, the results of these methods degrade in the crowded or complex scenes where there is a lot of occlusion.

Instead of trajectory features, local cuboid-based features are proposed to represent the events. These features include the histograms of gradients (HOG), histograms of optical flow (HOF), and other spatio-temporal gradient features that are extracted from local 2D image patches or local 3D video cuboids. For example, based on the SIFT (Scale-Invariant Feature Transform) features, Chen et al. [28] employed MoSIFT (Motion Scale Invariant Feature Transform), which can better describe motion intensity and has stronger discriminant power. Similarly, using MoSIFT descriptors, Xu [29] extracted the low-level features of the video to detect violent events. In order to take advantage of the global spatiotemporal distribution characteristics of interest points, Bregonzio et al. [15] accumulated interest points from multiple time dimensions to form an interest point cloud as global features for behavior recognition. Using densely sampled spatio-temporal video volumes (STVs), Roshtkhari [30] create both local and global compositional graphs of volumes at each pixel to represent the event. Some examples of these features are shown in Figure 1.

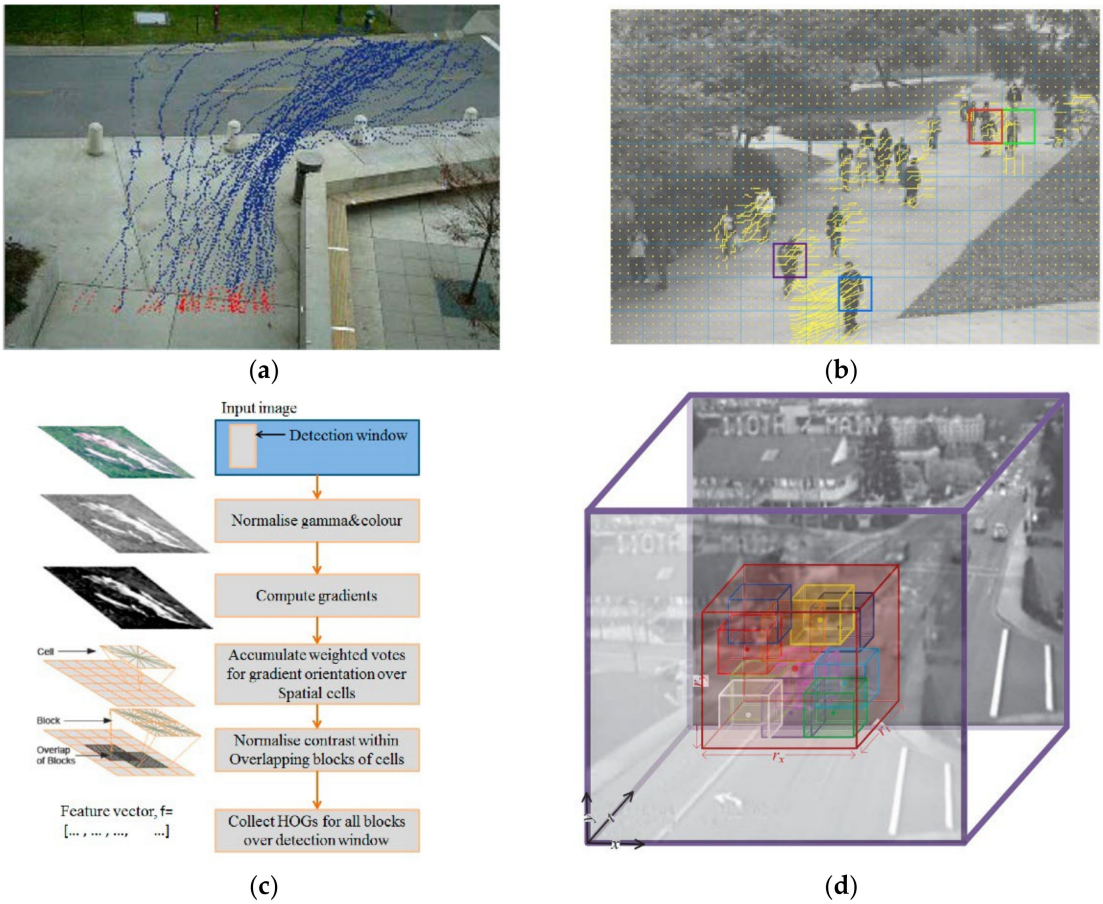


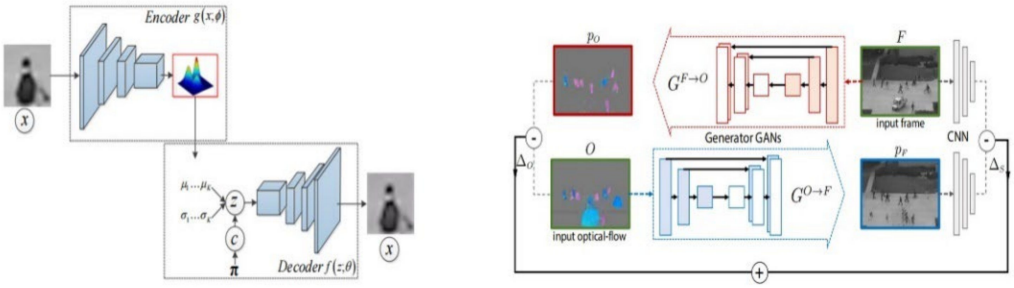
Figure 1. Examples of the hand-crafted feature. (**a**) Object trajectory [24]. (**b**) Dense trajectory [26]. (**c**) Histograms of gradients (HOG) [31]. (**d**) Spatio-temporal video volumes (STVs) [30].

However, it is quite difficult for hand-crafted-based methods to capture effective and robust behavior features due to the wide variety of monitoring scenes, complex crowd movement, and crowd density changes at any time, which can directly affect the anomaly-detection performance.

2.2. Deep Learning-Based Anomaly Detection Method

With the vigorous development of artificial intelligence technology, researchers began to explore detecting abnormal crowd behavior based on deep learning, which has yielded many results. Compared with the hand-crafted-based methods, the methods based on

deep learning focus on extracting the high-level features of pedestrian appearance and motion in the video and can further distinguish normal behavior from abnormal behavior. These methods include the technology that is based on the Convolutional Neural Network (CNN), Auto-Encoder and Generative Adversarial Network (GAN). It can be classified into two categories: (1) using the pre-trained CNN to extract features of the video frame to represent the event and to train a detection model with a one-class classifier [13]; (2) fusing with the RNN, optical flow information or 3D-CNN [14] to learn the regularity to detect the motion and appearance anomaly. The former method [13] achieved 90.8% frame-level AUC on the UCSD ped1 dataset, while the latter achieved 85.0% frame-level AUC on the UCSD ped2 dataset. The Auto-Encoder contains an encoder and a decoder and is mainly used for data dimensionality reduction and feature extraction. Given that video clips only contain normal events, the Auto-Encoder can reconstruct the normal event with a lower error while the abnormal event is constructed with a higher reconstruction error. Furthermore, the encoder could map the normal events to latent representations, by learning a detection model such as the Gaussian Mixture Model [15,32]. This method is called the Gaussian Mixture Fully Convolutional Variational Autoencoders (GMFC-VAE) and achieves 91.2% frame-level AUC on the UCSD ped2 dataset and 83.4% frame-level AUC on the Avenue dataset. The GAN [18,33] contains a generator and a discriminator, which can capture normal data probability and can estimate the probability that a sample fits the training data distribution. They achieved 93.5% frame-level AUC on the UCSD ped2 dataset and 99% frame-level AUC on the UMN dataset. Next, the reconstruction errors of the generator or the classify result of the discriminator are used to detect anomalies. Different from the method mentioned above, the proposed method called the Convolutional Recurrent AutoEncoder (CR-AE), which is an improved form of the CNN and Auto-Encoder, can capture the irregularity of the temporal pattern and spatial irregularity, respectively. Some examples of these deep learning-based method are shown in Figure 2.



(**a**) (**b**)

Figure 2. Examples of the deep learning-based method. (**a**) GMFC-VAE [32]. (**b**) GAN [18,33].

3. Method

Using the notation above, we formally introduce our approach in this section. We first state the anomaly detection problem formulation that we aim to deal with and then present the network architecture of the Convolutional Recurrent AutoEncoder (CR-AE).

3.1. Problem Formulation

The problem of the video event anomaly detection can be denoted as follows: In a video $\mathcal{V} = [C_i, i = 1, \ldots, T]$, where $C_i = [I_t, I_2, \cdots, I_{t+k-1}]$ represents the video clip of the frame, I_t and k are the length of the video frames. Here, the task is to assign each clip C_i a binary label to indicate whether this clip contains an anomaly event ($y_t = 1$) or not ($y_t = 0$).

An overview of the proposed method is illustrated in Figure 3. First, the video clips that only contain the normal event are used to learn the CR-AE network as the detection model. Then, the test video clip detects the anomaly or not by the reconstruction error.

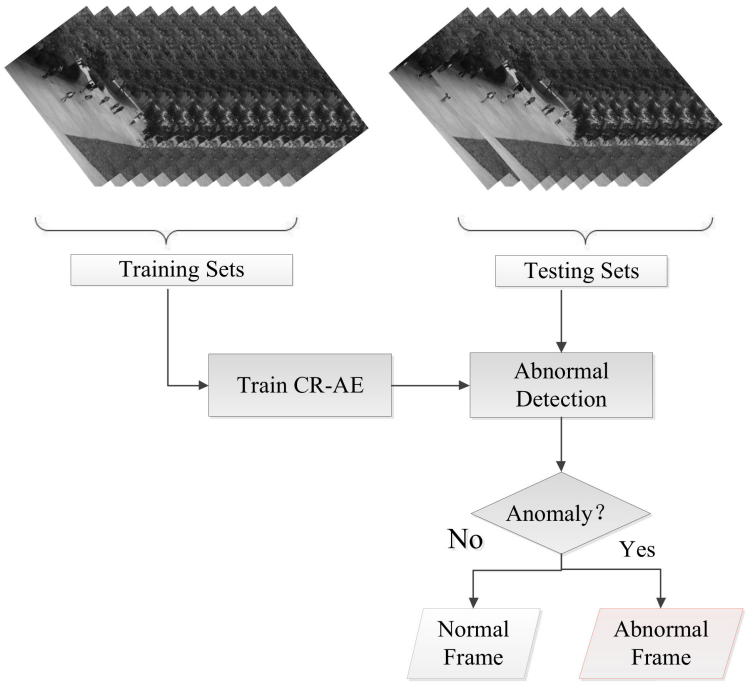


Figure 3. Overview of our proposed method.

3.2. Learning the CR-AE Network

The CR-AE network contains a Convolutional Encoder, an attention-based ConvLSTM and a Convolutional AutoEncoder. The encoder of the AutoEncoder is composed of multiple convolutional layers. In each layer of the encoder, the model first performs a convolution operation on the original input or the output of the previous layer and outputs the result of the convolutional layer to the Covn-LSTM layer. The attention mechanism is used to obtain the current output characteristics from the hidden state of each Covn-LSTM layer. In each layer of the Convolutional Decoder, the output feature of the previous decoder layer and the output feature of the encoder are merged, and they perform the deconvolution operation. Through layer-by-layer deconvolution, the input of the original video segment is reconstructed, and the 2-norm of the input and the reconstruction result are computed as the objective function. The architecture of the proposed CR-AE network is illustrated in Figure 4.

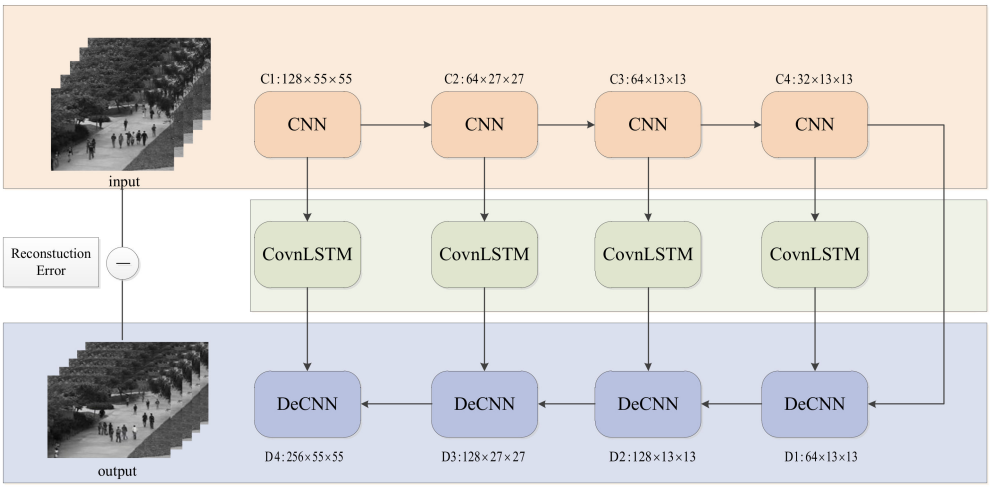


Figure 4. Overall architecture of the proposed CR-AE model.

In detail, the Convolutional Encoder encodes the input video clip. With the $(l-1)$-th layer feature maps $\mathcal{X}^{t,l-1} \in \Re^{n_{l-1} \times n_{l-1} \times d_{l-1}}$, the result of l-th layer can be represented as:

$$\mathcal{X}^{t,l} = f\left(W^l * \mathcal{X}^{t,l-1} + b^l\right) \tag{1}$$

where $*$ is the convolution operation and $f(\cdot)$ is the activation function. $W^l \in \Re^{k_l \times k_l \times d_{l-1} \times d_l}$ denotes d^l convolutional kernels of size $k_l \times k_l \times d_{l-1}$; $b^l \in \Re^{d_l}$ is a bias term, and $\mathcal{X}^{t,l} \in \Re^{n_l \times n_l \times d_l}$ is the output feature map at $l-th$ layer.

An attention-based ConvLSTM is adopted to capture the temporal regularity. By spanning different time steps, it can select hidden states, which are relevant to the last frames to overcome the deterioration of long-term dependencies. Furthermore, it can select relevant hidden states (feature maps) across different time steps to overcome the deterioration of the long-term dependence of the previous ConvLSTM [34]. Especially, with the l-th convolutional layer output feature $\mathcal{X}^{t,l} \in \Re^{n_l \times n_l \times d_l}$ of the Encoder from the previous hidden state $\mathcal{H}^{t-1,l} \in \Re^{n_l \times n_l \times d_l}$, the current hidden state $\mathcal{H}^{t,l}$ is updated with $\mathcal{H}^{t,l}$= ConvLSTM$(\mathcal{X}^{t,l}, \mathcal{H}^{t,l})$. Specifically, the detail of the ConvLSTM cell can be formulated as:

$$\mathbf{z}^{t,l} = \sigma\left(\widetilde{W}^l_{\mathcal{XZ}} * \mathcal{X}^{t,l} + \widetilde{W}^l_{\mathcal{HZ}} * \mathcal{X}^{t-1,l} + \widetilde{W}^k_{\mathcal{CZ}} \circ \mathcal{C}^{t-1,l} + \widetilde{b}^l_{\mathcal{Z}}\right) \tag{2}$$

$$\mathrm{r}^{t,l} = \sigma\left(\widetilde{W}^l_{\mathcal{XR}} * \mathcal{X}^{t,l} + \widetilde{W}^l_{\mathcal{HR}} * \mathcal{H}^{t-1,l} + \widetilde{W}^l_{\mathcal{CR}} \circ \mathcal{C}^{t-1,l} + \widetilde{b}^l_{R}\right) \tag{3}$$

$$\mathrm{C}^{t,l} = \mathbf{z}^{t,l} \circ \tanh\left(\widetilde{W}^l_{\mathcal{XC}} * \mathcal{X}^{t,l} + \widetilde{W}^l_{\mathcal{HC}} * \mathcal{H}^{t-1,l} + \widetilde{W}^l_{\mathcal{CR}} \circ \mathcal{C}^{t-1,l} + \widetilde{b}^l_{R}\right) + \mathbf{r}^{t,l} \circ \mathcal{C}^{t-1,l} \tag{4}$$

$$\mathbf{o}^{t,l} = \sigma\left(\widetilde{W}^l_{\mathcal{XO}} * \mathcal{X}^{t,l} + \widetilde{W}^l_{\mathcal{HO}} * \mathcal{H}^{t-1,l} + \widetilde{W}^l_{\mathcal{CO}} \circ \mathcal{C}^{t-1,l} + \widetilde{b}^l_{O}\right) \tag{5}$$

$$\mathcal{H}^{t,l} = \mathbf{o}^{t,l} \circ \tanh\left(C^{t,l}\right) \tag{6}$$

where $*$ and $\circ$ are the convolutional operator and hadamard product, respectively; σ is the activation function. $\widetilde{W}^l_{\mathcal{XZ}}$,$\widetilde{W}^l_{\mathcal{HZ}}$,$\widetilde{W}^k_{\mathcal{CZ}}$,$\widetilde{W}^l_{\mathcal{XR}}$,$\widetilde{W}^l_{\mathcal{HR}}$,$\widetilde{W}^l_{\mathcal{CR}}$,$\widetilde{W}^l_{\mathcal{XC}}$,$\widetilde{W}^l_{\mathcal{HC}}$,$\widetilde{W}^l_{\mathcal{XO}}$,$\widetilde{W}^l_{\mathcal{HO}}$,$\widetilde{W}^l_{\mathcal{CR}}$,$\widetilde{W}^l_{\mathcal{CO}}$ are the convolution kernels and $\widetilde{b}^l_{\mathcal{Z}}$,$\widetilde{b}^l_{R}$,$\widetilde{b}^l_{R}$,$\widetilde{b}^l_{\mathcal{O}}$ are the bias parameters. Different from the LSTM, all of the input, cell outputs, hidden states and gates are 3D tensors. The step length h is set as 5, and all the convolutional kernel sizes are set as the same. Next, a temporal attention mechanism is adopted to choose the relevant time steps and to obtain a refined output of feature maps $\mathcal{H}^{t,l}$, which can be expressed by:

$$\widehat{\mathcal{H}}^{t,l} = \sum_{i \in (t-h,t)} \alpha^i \mathcal{H}^{t,l} \tag{7}$$

$$\alpha^i = \frac{\exp\left\{\frac{V(\mathcal{H}^{t,l})V(\mathcal{H}^{i,l})}{\lambda}\right\}}{\sum_{i \in (t-h,t)} \exp\left\{\frac{V(\mathcal{H}^{t,l})V(\mathcal{H}^{i,l})}{\lambda}\right\}} \tag{8}$$

where $V(\cdot)$ denotes vector and λ is a rescale factor ($\lambda = 10.0$). The last hidden state $\mathcal{H}^{t,l}$ is used as a group-level context vector, and the importance weight α^i is measured by the softmax function. In this way, the attention-based ConvLSTM can capture the irregularity of the temporal pattern and spatial irregularity.

The Convolutional Decoder is used to decode the feature map obtained in the previous step to obtain the reconstructed video clip. In detail, the Convolutional Decoder is expressed as:

$$\mathcal{X}^{t,l-1} = \begin{cases} f\left(\widetilde{W}^{t,l}, \otimes, \widehat{\mathcal{H}}^{t,l}, +, b^{t,l}\right), \ l = 4 \\ f\left(\widetilde{W}^{t,l} \otimes \left(\widehat{\mathcal{H}}^{t,l} \oplus \widehat{\mathcal{X}}^{t,l}\right) + \widehat{b}^{t,l}\right), \ l = 3,2,1 \end{cases} \tag{9}$$

where $\otimes$ and $\oplus$ are the deconvolution and concatenation operations, respectively; $f(\cdot)$ is the activation unit; $\widetilde{W}^{t,l}$ and $b^{t,l}$ are the filter kernel and bias parameter, respectively. The reconstructed video clips from the previous layer of the decoder and the output of the previous ConvLSTM layer are combined and fed into the next deconvolution layer. The final output $\mathcal{X}^{t,0}$ denotes the reconstructed video clip.

The detailed configurations of the proposed CR-AE model architecture are presented in Section 4.3. Finally, the objective function of the CR-AE model can be defined as the reconstruction error over the input video clips as below:

$$\mathcal{L} = \sum_{k} \| \mathcal{V}_k - f_W(\mathcal{V}) \|_2^2 \tag{10}$$

where $\mathcal{V}_k$ and $f_W(\mathcal{V})$ are the video clip and reconstructed video clip.

3.3. Prediction

After training the model, the reconstruction error between the input frame $I^i_{x,y}$ and the reconstruction frame $f_w(I^i_{x,y})$ are represented as follows:

$$R(x,y,t) = \left\| I^i_{x,y} - f_w(I^i_{x,y}) \right\|_2 \tag{11}$$

where f_w is the learned CR-AE model. Then, the frame-level anomaly detection evaluation criteria can be represented by the sum of the all the pixel errors as below:

$$e(i) = \sum_{(x,y)} R(x,y,t) \tag{12}$$

Finally, the final frame-level score is

$$S_i = \frac{e(i) - \min_i e(i)}{\max_i e(i)} \tag{13}$$

The scores estimated from a frame of anomalous events are expected to be higher than those for normal events, and a threshold θ is set to determine the sensitivity of the anomalous detection method.

4. Experiment and Results

4.1. Datasets

To verify the method proposed in this paper, we performed experiments on two publicly available video anomaly datasets, namely the UCSD PED2 dataset [35] and the ShanghaiTech [36] dataset. Both of the two datasets have their own challenges and unique particularity, such as abnormal events, degradation of video quality, complex background environment, etc. Therefore, the model needs to be experimented on the two datasets separately, which are briefly introduced as follows.

The UCSD dataset is a collection of footage from a stationary camera overlooking the sidewalk at 10 frames per second. In this dataset, anomaly events are caused by non-pedestrians and abnormal pedestrian movements on the sidewalk. Specifically, some abnormal examples include cyclists, skaters, cars, etc. This dataset has two different subsets, PED1 and PED2, which are divided by the working direction. This paper only adopts the second scene, UCSD PED2 for experimentation. Ped2 is parallel to the camera plane and is split into 16 training clips and 14 test clips, consisting of 4560 frames and with a resolution of 320×240.

The ShanghaiTech dataset is one of the largest datasets and was created to expand scene diversity. Compared to the other dataset, the ShanghaiTech dataset contains more video clips, split into 330 training and 107 test video clips, which are taken in 13 different scenes and a large number of different anomaly types. There are around 316 K video frames

with a resolution of 856 × 480 in this dataset. In addition, it contains 130 abnormal events that include anomalies caused by sudden movements such as bicycles on the sidewalk, chases, and quarrels.

Figure 5 presents some examples of the two datasets.

(**a**) UCSD dataset Ped2

(**b**) ShanghaiTech dataset

Figure 5. This is a figure. Schemes follow the same formatting.

4.2. Implementation Details

Before training the model, many details need to be emphasized. We first convert all frames of the video clip to a grayscale image and then resize them to 227 × 227. Five consecutive frames are used as the input of the model. In detail, the C1-C4 consists of 128 3 × 3 convolutional kernels, 64 3 × 3 convolutional kernels, 64 3 × 3 convolutional kernels, and 32 3 × 3 convolutional kernels, as well as 2 × 2, 2 × 2, 2 × 2, and 2 × 2 strides, respectively. The Decoder comprises the reverse architecture of the encoder. It contains four deconvolutional layers: DeConv1-DeConv4 with 32 3 × 3 convolutional kernels, 64 3 × 3 convolutional kernels, 64 3 × 3 convolutional kernels, and 128 3 × 3 convolutional kernels, as well as 2 × 2, 2 × 2, 2 × 2, and 2 × 2 strides, respectively. The decoder can combine different deconvolutional and ConvLSTM layers to obtain the feature maps, which effectively improve the anomaly detection performance. Detail of the CR-AE model are shown in Table 1.

The network weights are initialized by the "Xavier" method and are optimized by the Adam optimizer [37] to minimize the above loss. The Adam optimizer computes dimensional learning rates to adjust the gradient rates through all previously updated functions in each dimension. The Adam optimizer is widely used due to its strong convergence and empirically successful theory. The learning rate of the Adam optimizer is set at a learning rate of 0.0001, a weight decay of 0:9 for each 100 epochs, a hyperparameter β_1 of 0.9, and a

hyperparameter β_2 of 0.999. The experiment is performed on a PC desktop with Intel Core i9-12900 CPU, NVIDIA GeForce GTX 3080 GPU and 32 GB RAM.

Table 1. Specifications of the CR-AE model.

Layer	Input	Kernel Size	Stride/ Pad	Output	Last/ Next Layer
Input	$5 \times 227 \times 227$				
Conv1	$5 \times 227 \times 227$	3×3	2/0	$128 \times 55 \times 55$	Input/Conv2 + Lstm1
Conv2	$128 \times 27 \times 27$	3×3	2/0	$65 \times 27 \times 27$	Conv 1/Conv3 + Lstm2
Conv3	$64 \times 27 \times 27$	3×3	2/0	$64 \times 13 \times 13$	Conv 2/Conv4 + Lstm3
Conv4	$64 \times 13 \times 13$	3×3	2/0	$32 \times 13 \times 13$	Conv 3/De-conv1 + Lstm4
Lstm1	$128 \times 55 \times 55$	N/A	N/A	$128 \times 55 \times 55$	Conv1/De-conv4
Lstm2	$64 \times 27 \times 27$	N/A	N/A	$64 \times 27 \times 27$	Conv2/De-conv3
Lstm3	$64 \times 13 \times 13$	N/A	N/A	$64 \times 13 \times 13$	Conv3/De-conv2
Lstm4	$32 \times 13 \times 13$	N/A	N/A	$32 \times 13 \times 13$	Conv4/De-conv1
De-conv1	$32 \times 13 \times 13$	3×3	2/0	$64 \times 13 \times 13$	Lstm4 + Conv4/De-conv2
De-conv2	$64 \times 13 \times 13$	3×3	2/0	$128 \times 27 \times 27$	Lstm3 + Conv1/De-conv3
De-conv3	$128 \times 27 \times 27$	3×3	2/0	$256 \times 55 \times 55$	Lstm2 + Conv2/De-conv4
De-conv4	$128 \times 55 \times 55$	3×3	2/0	$5 \times 277 \times 277$	Lstm3 + De-conv3/Output
Output	$5 \times 277 \times 277$				

Input, input layer; Conv, convolutional layer; Lstm, ConvLSTM layer; De-conv, deconvolutional layer; Output, output layer. The Encoder and Decoder consist of Conv1, Conv2, Conv3, Conv4 and De-conv1, De-conv2, De-conv3, De-conv4, respectively.

4.3. Results on the UCSD ped2 Dataset

On the UCSD ped2 dataset, we compared the results with the existing state-of-the-art methods, including the MPPCA [35], mixture dynamic texture (MDT) [35], 2D Convolutional AutoEncoder method (MT-FRCN [10], Conv2D-AE [14]), 3D Convolutional AutoEncoder method (Conv3D-AE) [14], AutoEncoder method based on Convolutional Long- and Short-term Memory Network (ConvLSTM-AE) [21], Stacked Recurrent Neural Network (StackRNN) [36], Baseline method [38], Semiparametric Scan Statistic (SSS) [39], Online GNG [40] and Unmasking [41], Appearance and Motion DeepNet (AMDN) [13]. Among these methods, the first five use handcrafted features and the last eight use deep learning techniques, the latter including common techniques such as convolutional neural networks, recurrent neural networks, autoencoders, generative adversarial networks, etc.

Frame-level evaluation criterion is adopted to evaluate the performance of the proposed method. For this criterion, the frame is determined as abnormal if at least one pixel of a frame is marked as abnormal. In order to use a frame-level criterion for evaluation, the time label is used to determine the true positive and false positive of the metric. Then, the detection rate (True Positive Rate, TPR) and false alarm rate (False Positive Rate, FPR) of the method are computed, as shown below:

$$TPR = \frac{TP}{TP + FN} \tag{14}$$

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

The receiver operating characteristics (ROC) are plotted with the true positive rate on the y-axis vs. the false positive rate. Then the area under the curve (AUC) is computed with different thresholds θ as the evaluation metric. A higher AUC score manifests better anomaly detection effects.

All the results of the comparison methods are taken from their respective papers. The qualitative frame-level evaluation results, in the form of ROC curves, are shown in Figure 6.

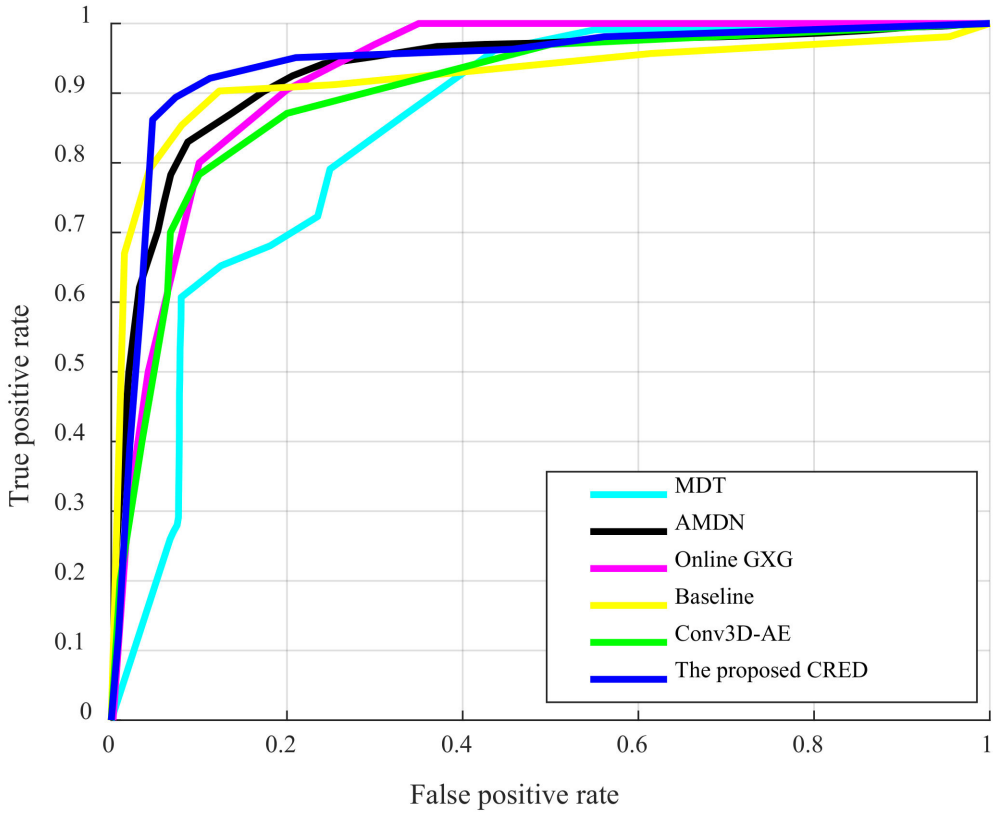


Figure 6. ROC curves for the UCSD Ped2 dataset.

From Figure 6, it can be observed that the proposed method achieves a larger area under the curve (AUC) except for the Baseline method [19]. By visual observation, it is difficult to distinguish the size of the AUC of these two methods from the figure. The quantitative results frame-level evaluation, in the form of AUC, are presented in Table 2. It is obvious from Table 2 that the AUC of the proposed method outperforms the Baseline method, with a 0.2% frame-level AUC lead. More specifically, the performance of the deep learning methods surpasses the hand-crafted features-based method. Among the fourteen algorithms, our method obtains the best result with a 95.6% frame-level AUC.

Table 2. Comparison with the state-of-the-art methods in terms of AUC% on the USCD Ped2 Dataset.

Method	AUC
MPPCA [35]	69.3%
MDT [35]	82.9%
SSS [39]	94.0%
Online GNG [40]	94.0%
Unmasking [41]	82.2%
ADMN [13]	90.8%
MT-FRCN [10]	92.2%
Conv2D-AE [14]	85.0%
Conv3D-AE [14]	91.2%
ConvLSTM-AE [21]	88.1%
StackRNN [36]	92.2%
Baseline [38]	95.4%
The proposed CR-AE	95.6%

4.4. Results on the ShanghaiTech Dataset

The ShanghaiTech dataset is a recently proposed dataset, has a large number of frames, and requires a relatively large calculation cost. Only a few methods have been tested

on this dataset. These compared methods include the Conv2D-AE [14], StackRNN [36], Baseline [38], Asymptotic Bound [32] and MemAE [23]. The quantitative results in the form of AUC are presented in Table 3.

Table 3. Comparison with the state-of-the-art methods in terms of AUC% on the ShanghaiTech dataset.

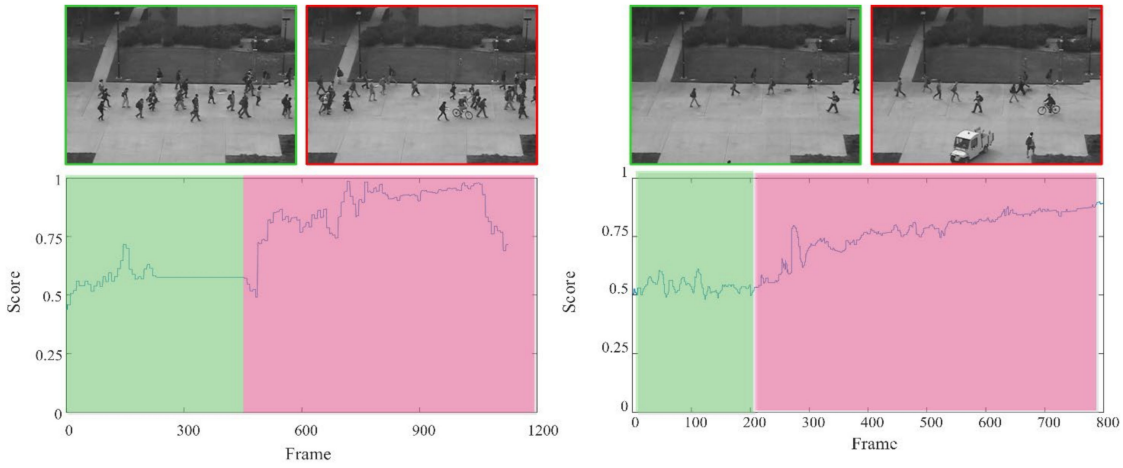
Method	AUC
Conv2D-AE [14]	60.9%
StackRNN [36]	68.0%
Baseline [38]	72.8%
Asymptotic Bound [32]	70.9%
MemAE [23]	71.2%
The proposed CR-AE	73.1%

It is shown that the proposed CR-AE method achieved the best detection results on this dataset. Specifically, the AUC of the proposed method is 0.3% better than the Baseline [38] method. However, the method proposed obtained a 73.1% frame-level AUC on the ShanghaiTech dataset, which is much lower than the frame-level AUC obtained on the UCSD Ped2 dataset. This is mainly because the ShanghaiTech dataset is more complex and contains more challenges. It is more complex, including multiple scenes, multiple frames, and abnormal events that have not previously appeared in other datasets.

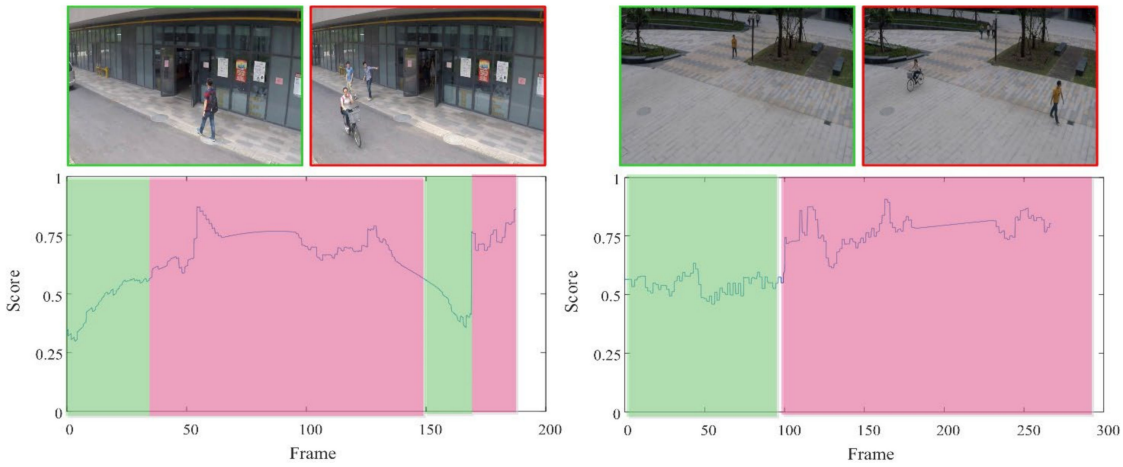
4.5. Visual Results

The detection results are visualized to further evaluate the performance of the proposed CR-AE model. As Figure 6 depicts, the frame-level detection results and some video screenshots of the two datasets are provided. In detail, the horizontal coordinate is the time of the video frame, the vertical coordinate abnormal score has been normalized to 1, and the red area represents the anomaly frames. It is evident that the proposed method can accurately detect video anomalies and can predict anomaly scores close to zero on normal videos, which demonstrates the effectiveness and robustness of the proposed CR-AE. Furthermore, it can be observed that the area with larger anomaly scores can correspond to the ground truth. Some abnormal events, such as bicycles and cars on the sidewalk, fights, and pushes, can basically be detected. Additionally, Figure 7 also provides some key frames of the detection results. When an abnormal event occurs suddenly, such as a car appearing on the scene in the right panel of Figure 7a, the anomaly score increases suddenly; on the contrary, if the anomalous object leaves the camera's field of view, as shown in the left panel of Figure 7b, the anomaly score rapidly declines.

Examples of better and worse abnormality detection results are shown in Figure 8. The first row shows the examples of the better cases with a higher frame-level score and, the second row shows the examples of the worse cases with a lower frame-level score. It is obvious that the anomalies such as cars (Figure 8a–d) and bikes (Figure 8b) move on the sidewalk, and intense movements (Figure 8c) are easy to detect. However, occluded (Figure 8e–g) and poorly illuminated (Figure 8f) anomalous objects are difficult to detect. Furthermore, detecting anomalous events with little movement such as a lost package (Figure 8h) is also challenging for the proposed methods.



(a) UCSD Ped2



(b) ShanghaiTech dataset.

Figure 7. Visualization of the testing results.

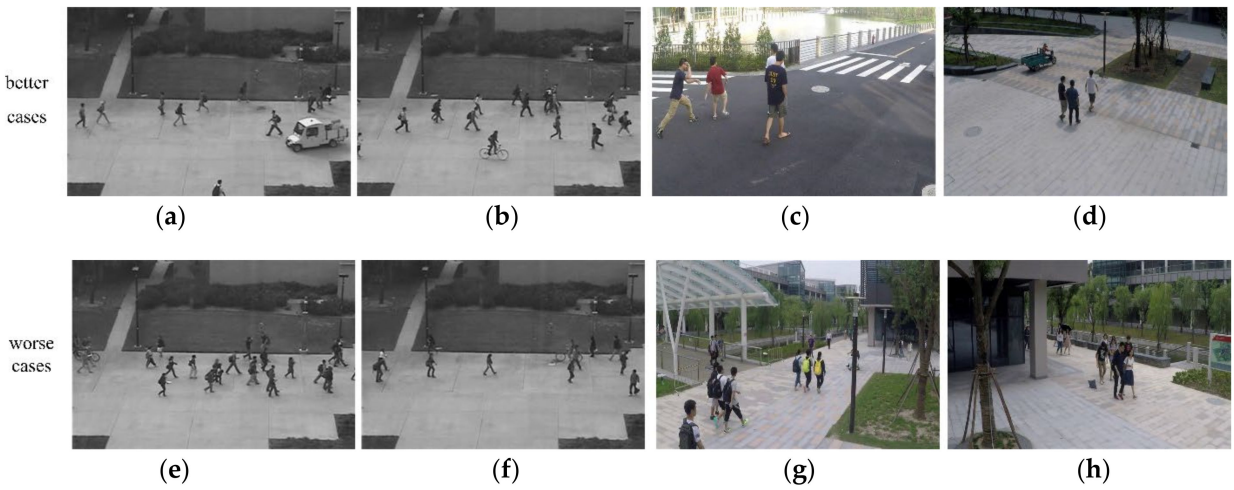


Figure 8. Examples of better and worse abnormality detection results. (**a**) cars on the sidewalk. (**b**) cyclists on the sidewalk. (**c**) intense movements. (**d**) cars on the sidewalk. (**e**) occluded, cyclists on the sidewalk (**f**) poorly illuminated, cyclists on the sidewalk. (**g**) occluded, scooters on the sidewalk. (**h**) lost package.

4.6. Computational Efficiency

Table 4 shows the detection speed comparison between our method and the other detection methods on the UCSD ped2 dataset, and the results of the comparison methods are from their corresponding articles. Some information, such as the computing environment and RAM, is not provided in these papers, but this does not affect the preliminary comparison of the results. The hardware environment of the whole experiment process is Intel Core i9-12900 CPU, NVIDIA GeForce GTX 3080 GPU and 32GB RAM, and the computing platform is Python 3.7 and Tensorflow 2.5. As can be seen from Table 3, the detection speed of the method proposed in this paper is 249 fps, which reaches the real-time detection speed (25 fps) and significantly exceeds the detection speed of other comparison methods.

Table 4. Running time comparison of the UCSD Ped2 dataset.

Method	Computing Environment	CPU	GPU	RAM	Detection Speed (fps)
MDT [35]	-	3.0 GHz	-	2.0 GB	0.04
StackRNN [36]	Python + Tensorflow	3.5 GHz	-	16 GB	120
AMDN [13]	MATLAB 2015	2.1 GHz	Nvidia Quadro K4000	32 GB	0.11
Unmasking [41]	Python + Tensorflow	-	GTX TITAN Xp	-	20
Proposed CR-AE	Python 3.7 + Tensorflow2.5	5.1 GHz	NVIDIA GTX 3080	32 GB	249

5. Conclusions

In this study, we have introduced a Convolutional Recurrent AutoEncoder (CR-AE) to explicitly model the normal dynamics in video sequences for anomaly detection. The framework was able to model both spatial and temporal irregularities of the video data, which are based on the combination of an attention-based Convolutional Long–Short-Term Memory (ConvLSTM) network and the convolutional encoder of the AutoEncoder. Then, the reconstruction errors of the convolutional decoder were further employed to detect abnormal frames. Both the qualitative and quantitative results showed that the proposed method outperforms the state-of-the-art anomaly detection method on the UCSD ped2 dataset and the ShanghaiTech dataset. In the future, we will further study online and adaptive model updating to improve the performance of video anomaly detection. The limitations of the study are that our method could only provide frame-level detection results, which are unable to locate anomaly events. Another future research focus is on object-level and pixel-level anomaly detection.

Author Contributions: B.W. proposed the main idea and wrote the paper; C.Y. revised the paper and improved the presentation. All authors have read and agreed to the published version of the manuscript.

Funding: This study received no funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Murugan, B.S.; Elhoseny, M.; Shankar, K.; Uthayakumar, J. Region-based scalable smart system for anomaly detection in pedestrian walkways. *Comput. Electr. Eng.* **2019**, *75*, 146–160. [CrossRef]
2. Park, H.; Noh, J.; Ham, B. A Survey on Deep Learning Techniques for Video Anomaly Detection. *Int. J. Comput. Eng. Inf. Technol.* **2021**, *10*, 184–191.
3. Luo, W.; Liu, W.; Lian, D.; Tang, J.; Duan, L.; Peng, X.; Gao, S. Video anomaly detection with sparse coding inspired deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1070–1084. [CrossRef] [PubMed]

4. Cosar, S.; Donatiello, G.; Bogorny, V.; Garate, C.; Aivares, O.; Bemond, F. Toward abnormal trajectory and event detection in video surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *27*, 683–695. [CrossRef]
5. Mansour, R.F.; Gutierrez, J.E.; Gamarra, M.; Villanueva, J.A.; Leal, N. Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model. *Image Vis. Comput.* **2021**, *112*, 104229. [CrossRef]
6. Fernando, T.; Gammulle, H.; Denman, S. Deep Learning for Medical Anomaly Detection A Survey. *ACM Comput. Surv.* **2022**, *54*, 1–37. [CrossRef]
7. Xu, K.; Jiang, X.; Sun, T. Anomaly detection based on stacked sparse coding with intraframe classification strategy. *IEEE Trans. Multimed.* **2018**, *20*, 1062–1074. [CrossRef]
8. Cheng, K.; Chen, Y.; Fang, W. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2909–2917.
9. Zhang, X.; Yang, S.J.; Zhang, W. Video anomaly detection and localization using motion-field shape description and homogeneity testing. *Pattern Recognit.* **2020**, *105*, 107394. [CrossRef]
10. Leyva, R.; Sanchez, V.; Li, C. Video anomaly detection with compact feature sets for online performance. *IEEE Trans. Image Process.* **2017**, *26*, 3463–3478. [CrossRef] [PubMed]
11. Huang, L.; Cao, L.; Li, N. A state perception method for infrared dim and small targets with deep learning. *Chin. Opt.* **2020**, *13*, 527–536.
12. Ionescu, R.; Georgescu, M.; Shao, L. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7834–7843.
13. Xu, D.; Yan, Y.; Ricci, E.; Sebe, N. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Comput. Vis. Image Underst.* **2017**, *156*, 117–127. [CrossRef]
14. Hasan, M.; Choi, J.; Neumanny, J. Learning temporal regularity in video sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 7–12 June 2016; pp. 733–742.
15. Fan, Y.; Wen, G.; Li, D.; Qiu, S.; Levine, M.D. Video Anomaly Detection and Localization via Gaussian Mixture Fully Convolutional Variational Autoencoder. *Comput. Vis. Image Underst.* **2020**, *195*, 102920. [CrossRef]
16. Hinton, G.; Salakhutdinov, R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
17. Ramachandra, B.; Jones, M.; Vatsavai, R. Learning a distance function with a Siamese network to localize anomalies in videos. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 2–5 March 2020; pp. 2587–2596.
18. Ravanbakhsh, M.; Nabi, M.; Mousavi, H.; Sangineto, E.; Sebe, N. Abnormal event detection in videos using generative adversarial nets. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 1577–1581.
19. Ruff, L.; Vandermulen, R.; Gornitz, N.; Deecke, L.; Siddiqui, S.; Binder, A.; Muller, E.; Kloft, M. Deep One-Class Classification. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2019; pp. 6981–6996.
20. Abati, D.; Porrello, A.; Calderara, S.; Cucchiara, R. Latent space autoregression for novelty detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 481–490.
21. Luo, W.; Liu, W.; Gao, S. A revisit of sparse coding based anomaly detection in stacked rnn framework. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 341–349.
22. Teed, Z.; Deng, J. RAFT: Recurrent All Pairs Field Transforms for Optical Flow. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 4839–4843.
23. Dong, G.; Liu, L.; Vuong, L.; Budhaditya, S.; Moussa, M.R.; Svetha, V.; Hengel, A. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the IEEE International Conference on Computer Vision, Los Angeles, CA, USA, 15–21 June 2019; pp. 1705–1714.
24. Junejo, I. Using dynamic Bayesian network for scene modeling and anomaly detection. *Signal Image Video Process.* **2010**, *4*, 1–10. [CrossRef]
25. Sekh, A.; Dogra, D.; Kar, S. Video trajectory analysis using unsupervised clustering and multi-criteria ranking. *Soft Comput.* **2020**, *24*, 16643–16654. [CrossRef]
26. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.-L. Dense trajectories and motion boundary descriptors for action recognition. *Int. J. Comput. Vis.* **2013**, *103*, 60–79. [CrossRef]
27. Wang, H.; Schmid, C. Action recognition with improved trajectories. In Proceedings of the IEEE International Conference on Computer Vision, San Francisco, CA, USA, 15–17 June 2013; pp. 3551–3558.
28. Convertini, N.; Dentamaro, V.; Impedovo, D. A controlled benchmark of video violence detection techniques. *Information* **2020**, *11*, 321. [CrossRef]
29. Xu, L.; Gong, C.; Yang, J. Violent video detection based on MoSIFT feature and sparse coding. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 4–9 May 2014; pp. 3538–3542.
30. Roshtkhari, M.J.; Levine, M.D. Online dominant and anomalous behavior detection in videos. In Proceedings of the IEEE International Conference on Computer Vision, Portland, OR, USA, 7–13 June 2013; pp. 2611–2618.
31. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 5–11 June 2005; pp. 886–893.
32. Keval, D.; Yasin, Y. Online Anomaly Detection in Surveillance Videos with Asymptotic Bounds on False Alarm Rate. *Pattern Recognit.* **2021**, *114*, 107865.
33. Song, H.; Sun, C.; Wu, X. Learning Normal Patterns via Adversarial Attention-Based Autoencoder for Abnormal Event Detection in Videos. *IEEE Trans. Multimed.* **2020**, *22*, 2138–2148. [CrossRef]

34. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W.-C. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In Proceedings of the Conference and Workshop on Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 802–810.
35. Mahadevan, V.; Li, W.; Bhalodia, V.; Vasconcelos, N. Anomaly detection in crowded scenes. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 15–17 June 2010; pp. 1975–1981.
36. Luo, W.; Liu, W.; Gao, S. Remembering history with convolutional lstm for anomaly detection. In Proceedings of the IEEE International Conference on Multimedia and Expo, Hong Kong, China, 10–14 July 2017; pp. 439–444.
37. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations, Bejing, China, 22–24 June 2014.
38. Liu, W.; Luo, W.; Lian, D.; Gao, S. Future frame prediction for anomaly detection—A new baseline. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6536–6545.
39. Hu, Y.; Zhang, Y.; Davis, L. Unsupervised Abnormal Crowd Activity Detection Using Semiparametric Scan Statistic. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, San Francisco, CA, USA, 15–17 June 2013; pp. 767–774.
40. Sun, Q.; Liu, H.; Harada, T. Online Growing Neural Gas for Anomaly Detection in Changing Surveillance Scenes. *Pattern Recognit.* **2017**, *64*, 187–201. [CrossRef]
41. Ionescu, R.; Smeureanu, S.; Alexe, B.; Popescu, M. Unmasking the Abnormal Events in Video. In Proceedings of the International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2914–2922.