applied sciences

MDPI

Article

Improved Video Anomaly Detection with Dual Generators and Channel Attention

Xiaosha Qi [1], Zesheng Hu [2] and Genlin Ji [2,*]

[1] School of Mathematical Sciences, Nanjing Normal University, Nanjing 210023, China
[2] School of Computer and Electronic Information/Artificial Intelligence, Nanjing Normal University, Nanjing 210023, China
* Correspondence: glji@njnu.edu.cn

Abstract: Video anomaly detection is a crucial aspect of understanding surveillance videos in real-world scenarios and has been gaining attention in the computer vision community. However, a significant challenge is that the training data only include normal events, making it difficult for models to learn abnormal patterns. To address this issue, we propose a novel dual-generator generative adversarial network method that improves the model's ability to detect unknown anomalies by learning the anomaly distribution in advance. Our approach consists of a noise generator and a reconstruction generator, where the former focuses on generating pseudo-anomaly frames and the latter aims to comprehensively learn the distribution of normal video frames. Furthermore, the integration of a second-order channel attention module enhances the learning capacity of the model. Experiments on two popular datasets demonstrate the superiority of our proposed method and show that it can effectively detect abnormal frames after learning the pseudo-anomaly distribution in advance.

Keywords: generative adversarial networks; video anomaly detection; reconstruction

check for updates

Citation: Qi, X.; Hu, Z.; Ji, G. Improved Video Anomaly Detection with Dual Generators and Channel Attention. *Appl. Sci.* **2023**, *13*, 2284. https://doi.org/10.3390/app13042284

Academic Editor: David Megías

Received: 21 December 2022
Revised: 4 February 2023
Accepted: 8 February 2023
Published: 10 February 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The proliferation of surveillance videos has driven the advances in video anomaly detection, which focuses on mining unusual patterns from these videos. In video anomaly detection, anomalies refer to events or behaviors that differ from normal scenes. Anomalies can take many forms in different environments:

- Abnormal human behaviors, i.e., inappropriate behavior in public places, e.g., climbing, filming, or stealing, etc.
- Vehicle abnormality, such as illegal parking, violating traffic rules, etc.
- Abnormal environments, e.g., fire, explosion, building collapse, etc.
- Equipment failure, such as surveillance camera failure, light changes, etc.

These anomalies are the targets that are to be detected and identified by the video anomaly detection system. They are of great significance for security monitoring, traffic management, emergency management, and other fields.

The determination of whether an event in a video is abnormal depends on the specific context of the scene. For instance, a truck might be considered abnormal on a university campus, but normal on a busy road. The problem of identifying anomalies arises from the vast variety of such events, which makes it challenging to gather enough training data to cover all possibilities. As a result, anomaly detection models often only learn to recognize normal activities that occur within a particular scene.

Manually screening these videos is a labor-intensive process, so deep learning methods have become increasingly popular in video anomaly detection tasks [1]. Initially, these tasks utilized traditional machine learning techniques that relied on manual feature extraction and model building. However, with the advent and ongoing advancement of deep learning,

video anomaly detection has evolved into a hybrid approach that combines both traditional machine learning and deep learning. The majority of current methods detect anomalies by extracting deep features and incorporating traditional machine learning techniques.

In recent years, we have seen a proliferation of deep learning-based methods that have been continuously improved and refined. These methods can extract features from videos more effectively and efficiently and are better equipped to learn the normal distribution. Common deep learning-based video anomaly detection methods include reconstruction and prediction. In this paper, we propose a novel method of detecting anomalies by reconstructing video frames and calculating the reconstruction error, as demonstrated in Figure 1.

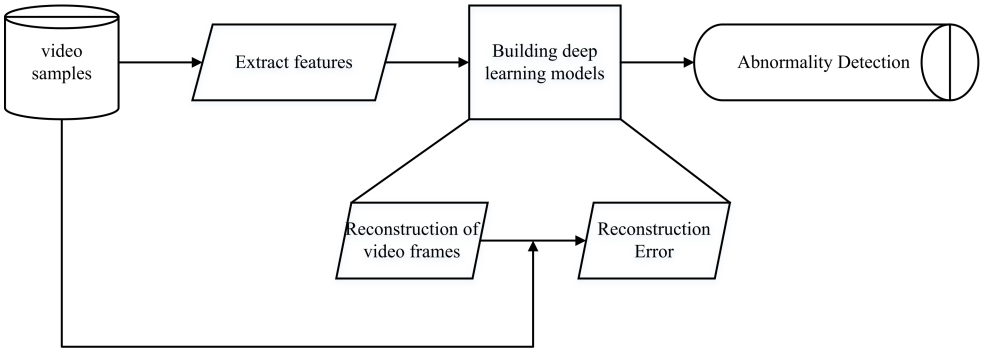


Figure 1. Illustration of the reconstruction-based video anomaly detection method. The video dataset is divided into individual frames, which undergo feature extraction to produce a feature map. During the training phase, the normal video frames are used to train the model for reconstruction. In the testing phase, the test video frames are fed into the trained model for reconstruction, and the distance between the reconstructed features and the original features is calculated and normalized to obtain the reconstruction error. The reconstruction error is then compared against a predetermined threshold to classify whether the video frame is normal or anomalous.

In the field of video anomaly detection, autoencoders are commonly used deep learning networks for reconstructing video frames and extracting their latent features. However, autoencoders tend to result in blurry edges during reconstruction. To address this issue, we introduce the use of generative adversarial networks (GAN) as the main model structure, with the autoencoder serving as the generator. This ensures stable network training and improves the clarity of the reconstructed frames. To further enhance feature information relevance, we add a second-order channel attention module to the autoencoder, which learns the internal dependencies of features through second-order feature distributions, resulting in improved network focus on beneficial information and enhanced classification ability.

Current methods often rely on training sets solely comprised of normal events, which can result in the model reconstructing abnormal frames during testing due to its strong reconstruction capabilities. To tackle this problem, we add another generator to the GAN, which transforms normal training frames into pseudo-abnormal frames [2]. This supplements the lack of abnormal frames in the original training set and enables the model to learn the pattern of abnormal events in advance. Our proposed method, referred to as the dual-generator generative adversarial network (DGGAN), features a dual-generator structure and incorporates GAN. The main contribution of our work consists of three parts:

- A dual-generator generative adversarial network (DGGAN) is proposed to improve the accuracy of video anomaly detection;
- A noise generator is designed to generate pseudo-anomaly frames to train the model, which improves the ability of the model to perceive unknown anomalies;
- A second-order channel attention module is used to learn feature interdependencies to better utilize the important feature information.

The rest of the article structure is presented as follows. Section 2 briefly describes the recent related work. Section 3 details the components of the proposed DGGAN. Section 4 shows the superiority of DGGAN through experiments and comparisons. Finally, we summarize the work and look forward to promising research directions.

2. Related Work

The methods for video anomaly detection have evolved from traditional statistical inference techniques to cutting-edge deep learning models, including reconstruction and prediction approaches. These deep learning-based methods enhance both the quality of video frame feature extraction and the accuracy of detection [3–6]. The widely used techniques include autoencoders, generative adversarial networks, and attention mechanisms.

The autoencoder network is designed to uncover the hidden features of input data and obtain more effective features. The network comprises an encoder and a decoder. The encoder reduces the dimensionality of the input video frame and extracts the global latent features associated with it. The decoder then uses the learned Gaussian distribution to upscale the extracted latent features, producing a map that resembles the original video frame. Autoencoders are typically successful in accurately reconstructing anomalies, even when trained on normal data alone. However, this capability can sometimes lead to a decline in detection performance [7,8]. To this end, Reference [9] proposes spatial rotation transformation and temporal mixing transformation to avoid the generative ability of the model to even predict anomalies. References [10,11] propose to employ a memory mechanism on the latent space between the encoder and decoder of an autoencoder to limit the reconstruction ability in the case of abnormal inputs. Similarly, Reference [12] proposes to train an unsupervised object-centric convolutional autoencoder to extract feature vectors, divide the training samples into different clusters, use SVM to divide different clusters, and perform anomaly scoring for the classification.

Generative adversarial networks (GANs) [13] consist of two key components: generators and discriminators. These elements engage in an adversarial learning process where the generator produces fake data samples and tries to trick the discriminator, while the discriminator attempts to identify the generated samples as different from the actual data. Through repeated cycles of competition, both the generator and discriminator continually improve their abilities [14]. For instance, Reference [15] proposes a context-related video anomaly detection method combined with a two-branch generative adversarial network. Reference [16] presents a generative adversarial network with a dual discriminator to predict future frames.

The amount of information a model can store, as well as its expressive power, grows with the parameters. However, the large number of parameters can lead to information overload. The attention mechanism [17] enables the model to focus on the information that is more critical to the current task, alleviating the problem of information overload and model inefficiency [18–20]. To make the extractor pay more attention to abnormal regions during feature extraction, Reference [21] proposes an extractor augmented by a self-guided attention module. Reference [22] proposes a residual attention-based autoencoder for video anomaly detection.

In this paper, we present a novel approach to video anomaly detection that leverages the power of GAN and integrates attention mechanisms and autoencoders. Concretely, our method is built on the general framework of GANs and employs two generators to produce pseudo frames and reconstructed frames. The generator incorporates autoencoders to ensure stable training, and the attention mechanism is embedded to enhance the interdependence among the video frames and to concentrate on vital feature information. The details of our work will be described in detail in Section 3.

3. Method

3.1. Overall Framework

The framework of dual-generator generative adversarial networks (DGGANs) is shown in Figure 2. The method is mainly divided into two stages, namely the training and testing stages. The training phase starts after obtaining the video frames split by the video training set.

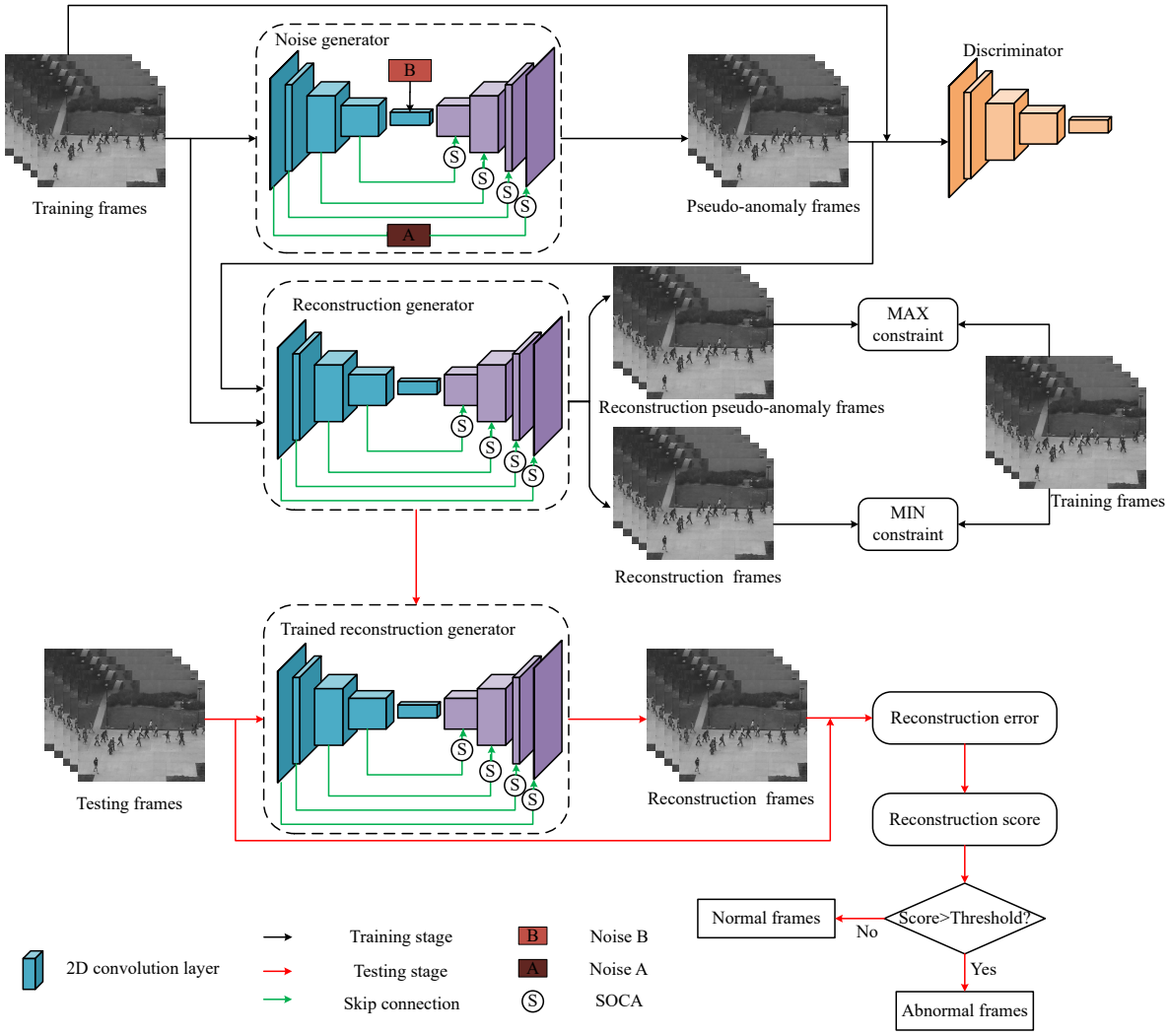


Figure 2. The overall pipeline of DGGAN. In the training stage, the noise generator and the reconstruction generator are trained independently. The former generates pseudo-abnormal frames using the training data, while the latter strives to improve its capacity for reconstructing normal frames by being trained on both the training data and pseudo-abnormal frames. During the testing phase, the testing data are fed into the trained reconstruction generator, and the resulting reconstruction score is used to classify if the data are abnormal or not.

In the first step of the process, the noise generator is trained to generate pseudo-abnormal video frames from the normal video frames. This is achieved by training the generator against the discriminator so that the generated frames cannot be distinguished as anomalies by the discriminator.

Next, the parameters of the generator are fixed, and the noise module is combined with the generator module. The generator is then confronted with the discriminator again, and the parameters of the noise module are updated to obtain the noise generator. The reconstruction generator is then trained, with the abnormal pattern learned in advance through the pseudo-abnormal frames generated by the noise generator. The generated

pseudo-abnormal frames and real frames are then input into the reconstruction generator to obtain reconstructed pseudo-abnormal frames and reconstructed real frames. The constraints function is added between the reconstructed frame and the real frame, so that the difference between the reconstructed pseudo-abnormal frame and the real frame is maximized, while the difference between the reconstructed real frame and the real frame is minimized.

Finally, the trained reconstruction generator is applied to the video frames of the testing set. The test frames are input into the reconstruction generator to obtain the reconstructed frames. The reconstruction error between the reconstructed frames and the real frames is calculated and used to classify the video frames in sequence.

3.2. Components

DGGAN is mainly composed of the noise generator, reconstruction generator, and discriminator.

3.2.1. Noise Generator

The noise generator (Figure 3) exists only in the training phase of our method and consists of a generator, two noise modules, and a second-order channel attention module.

The generator network in this process is constructed using an autoencoder, which consists of both an encoder and a decoder. When the training frame is input into the generator, it passes through the encoder, made up of pooling layers, convolution layers, and activation functions, which repeatedly halves the size and doubles the channels. Eventually, the encoder produces latent features that are fed into the second-order channel attention module to improve the interdependence of the features, enrich the information correlation, and produce features that reflect both global and local aspects. The decoder then reconstructs the latent features, producing a high-quality reconstructed frame that is the same size as the original input. The reconstructed frame is then compared to the real frame by the discriminator, which judges whether the reconstructed frame is similar enough to the real frame to be considered real.

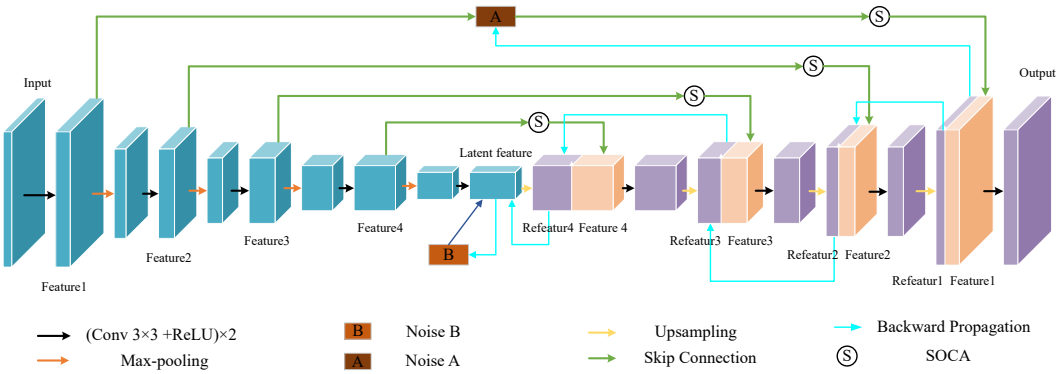


Figure 3. Illustration of the noise generator. The generator undergoes two training iterations. The first time is to train the generator without the noise module, and the second time is to train the noise generator with the noise module added. The noise generator is mainly used in the training phase to generate pseudo-anomaly frames, but not in the testing phase.

To generate pseudo-abnormal video frames, the generator network has been augmented with two noise modules, which consist of a three-layer fully-connected autoencoder. Random noise is input into the noise module, which processes the noise through cubic convolution, batch normalization, and activation functions to produce noise that can be used to generate fake video frames. The generator network and noise module are then combined, with a skip connection added between the first convolution pooling of the encoder and the last upsampling of the decoder, and the noise module A added. Additionally, noise module B is added after obtaining the latent features, and the latent features are reconstructed

with the added noise to produce pseudo-abnormal frames. Both noise modules have the same structure (as shown in Figure 4), but the input feature dimensions differ. Finally, the pseudo-abnormal frame and real frame are input into the discriminator, and the generator and discriminator compete to maximize the distance between the two frames.

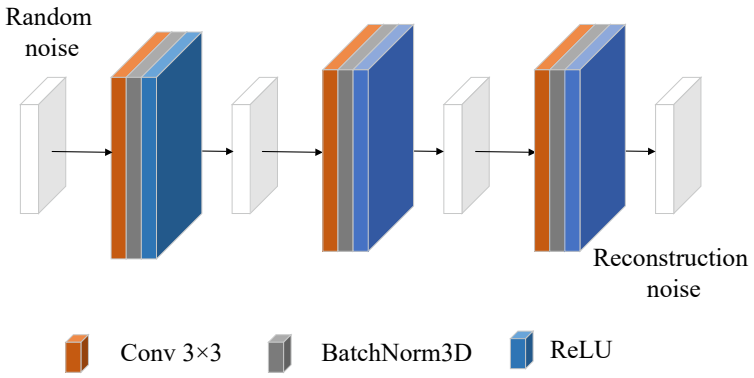


Figure 4. Illustration of the noise module. After inputting random noise, the noise module reconstructs it to obtain noise suitable for generating pseudo-frames. The module consists of a three-layer fully connected structure. Among them, noise A is added at the first skip connection, and noise B is added to the latent feature, and the size of both is the same as the corresponding feature size. "Conv 3 × 3" refers to using a convolution kernel of size 3 × 3. "BatchNorm3D" is a three-dimensional batch normalization used to normalize the input so that the input of each layer has the same data distribution. "ReLU" stands for using the ReLU activation function.

3.2.2. Reconstruction Generator

The reconstruction generator (Figure 5) is composed of an autoencoder and a second-order channel attention module. The autoencoder consists of an encoder and a decoder, and U-net is used as its network structure.

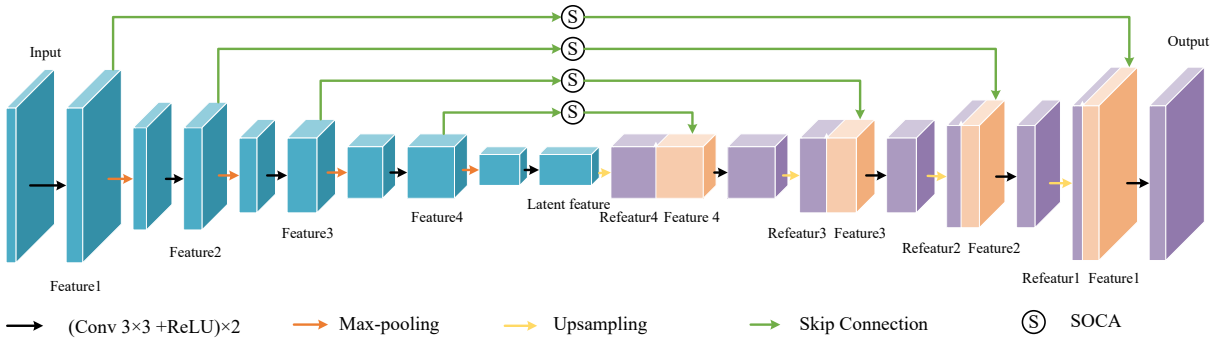


Figure 5. Illustration of the reconstruction generator. During the training phase, the generator is trained with both pseudo-abnormal frames and normal training frames to enhance its reconstruction capability. During the testing phase, the generator classifies input frames as abnormal or normal based on the distance between the reconstructed test frame and the input test frame.

During the training phase, the generator is trained using both pseudo-abnormal and real frames. When a frame is input into the generator, it first goes through the encoder module, which includes a series of convolution pooling operations. This results in the extraction of latent features of varying sizes and channels from each layer. Next, the second-order channel attention module processes these latent features to enhance the interdependence and relationships between the information. The resulting features, which have stronger correlations, are then decoded by the decoder module through up-sampling and skip connections to produce a reconstructed frame of the same size as the input frame.

Finally, constraints are applied to maximize the difference between the reconstructed pseudo-abnormal frame and the real frame, and minimize the difference between the

reconstructed real frame and the real frame. During the testing phase, the generator uses the distance between the reconstructed test frame and the input test frame to classify the input frame as either normal or abnormal.

3.2.3. Second-Order Channel Attention Module

The second-order channel attention (SOCA) [23] module explores the attention of second-order feature statistics based on the first-order channel attention module. The global covariance pooling operation is adopted, and the Newton iteration method is used to solve the covariance normalization to reduce the required computational resources. As shown in Figure 6, after the feature map is input into the second-order channel attention module, the global covariance pooling operation is first performed. Reshape the input feature to a matrix X of $C \times H \times W$, and calculate its corresponding covariance matrix $\sum$. $\sum$ can be obtained by Formulas (1) and (2):

$$\sum = X\bar{I}X^{\mathrm{T}} \tag{1}$$

$$\bar{I} = \frac{1}{s}(I - \frac{1}{s}O) \tag{2}$$

where I and O represent the $s \times s$ identity matrix and all-one matrix, respectively.

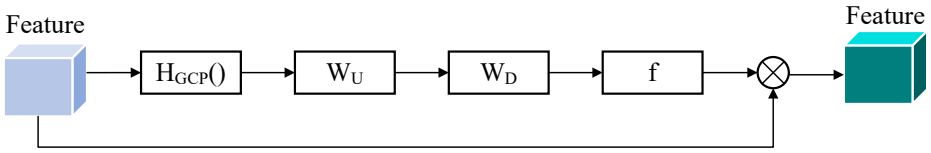


Figure 6. Illustration of the SOCA.

We use $\sum$ for covariance normalization. Since this matrix is a symmetric positive definite matrix, its eigenvalue decomposition is shown in Formula (3):

$$\sum = U\Lambda U^{\mathrm{T}} \tag{3}$$

where U is an orthogonal matrix, and Λ refers to the diagonal matrix when the eigenvalues are not increasing. The covariance normalization is converted to the power of the eigenvalue, as shown in Formula (4):

$$\hat{Y} = \sum^{\alpha} = U\Lambda^{\alpha}U^{\mathrm{T}} \tag{4}$$

where α represents a positive real number, and Λ^{α} represents a diagonal matrix. When $\alpha = 1$, no normalization is required, and when $\alpha < 1$, it will nonlinearly shrink eigenvalues greater than 1.0. Given previous work [24], we adopt $\alpha = 0.5$.

The processed features go into SOCA. After two layers of convolution, the first layer of convolution is used for channel dimensionality reduction and the second layer of convolution is used for channel restoration, and the final second-order channel attention vector ω is obtained. ω can be obtained by Formula (5):

$$\omega = f(W_U\delta(W_D z)) \tag{5}$$

where W_d and W_u are the weights of the convolutional layer, the feature channels are C/r and C, $f(\cdot)$ represents the sigmoid function, and δ represents the ReLu function. Let $\hat{Y} = [y_1, \cdots, y_c]$, the channel-wise statistics $z \in R^{C\times 1}$ can be obtained by shrinking $\hat{Y}$. Then the c-th dimension of z is computed as Formula (6):

$$z_c = H_{GCP}(y_c) = \frac{1}{C}\sum_{i}^{C} y_c(i) \tag{6}$$

where $H_{GCP}(\cdot)$ represents the global covariance pooling function. Compared with first-order pooling, global covariance pooling can obtain higher-order feature information and more discriminative feature information.

Finally, the second-order channel attention vector and the input features are multiplied by channel to obtain new features related to internal information.

3.3. Constraint Function

In order for the network to converge better during training, we use a constraint function to constrain the network. We constrain the appearance and motion levels to make the pseudo-anomaly frames generated by the noise generator far away from the real frames. Meanwhile, the distance between the reconstructed pseudo-abnormal frame generated by the reconstruction generator is far from the real frame, and the distance between the reconstructed real frame and the real frame is close. The apparent constraints L_{app} are divided into gradient constraints L_{gc} and strength constraints L_{sc}. Here, L_{gc} and L_{sc} can be obtained by Formulas (7) and (8), respectively:

$$\begin{aligned} L_{gc}(\hat{x}, x) = \sum_{a,b} (\| |\hat{x}_{a,b} - \hat{x}_{a-1,b}| - |x_{a,b} - x_{a-1,b}| \|_1 \\ + \| |\hat{x}_{a,b} - \hat{x}_{a,b-1}| - |x_{a,b} - x_{a,b-1}| \|_1) \end{aligned} \tag{7}$$

$$L_{sc}(\hat{x}, x) = \|\hat{x} - x\|_2^2 \tag{8}$$

where x is the input original frame, $\hat{x}$ is the reconstruction frame, a and b denote the horizontal and vertical coordinates of the frame pixels.

Then the apparent constraints L_{app} can be calculated from the gradient constraints L_{gc} and strength constraints L_{sc}. L_{app} can be calculated by Formula (9):

$$L_{app} = mL_{gc} + nL_{sc} \tag{9}$$

where $m : n = 1 : 1$.

The motion constraints L_{opt} are optical flow constraints [25], and the calculation process is shown in Formula (10):

$$L_{opt} = \|f(\hat{x}_{t+1}, x_t) - f(x_{t+1}, x_t)\|_1 \tag{10}$$

where t represents the t-th frame.

In the training noise generator phase, the generator of the noiseless module is first trained. To shorten the distance between the reconstructed frame and the real frame, the discriminator is fixed, and the generator is constrained at the apparent level. The objective function G_o is shown in Formula (11):

$$G_o = \min_{\theta} \|\hat{x} - x\|_2^2 \tag{11}$$

Then we train the corresponding discriminator and fix the generator. The discriminator can classify the reconstructed frames as abnormal frames and the real frames as normal frames, to accurately classify the reconstructed frames and the real frames. After label smoothing [26], label outlier 0 represents normal and label outlier 1 represents abnormal; they are replaced by 0.05 and 0.95, respectively. The loss function L_{adv}^D is shown in Formula (12):

$$L_{adv}^D(\hat{x}, x) = \frac{1}{2} \sum_{i,j} [L(D(x)_{i,j}, 0.95) + L(D(\hat{x})_{i,j}, 0.05)] \tag{12}$$

where i, j are the indices of the frame, $D(\cdot) \in [0, 1]$, $L(\cdot, \cdot)$ denotes the absolute value of the difference between the two.

After obtaining the trained generator and discriminator, we add a noise module to the generator, then fix the discriminator, and add a constraint function at the apparent level to widen the distance between the generated frame and the real frame. We update the

parameters of the noise module, and make the noise generator generate pseudo-anomaly frames, and its objective function L_n is shown in Formula (13):

$$L_n = \max_{\theta_n} \|G_o(x;\theta_n) - x\|_2^2 \tag{13}$$

where θ_n is the parameter of the noise module.

In the stage of training the reconstruction generator, the maximum and minimum constraints are mainly applied to the reconstruction generator in terms of appearance and motion. For the pseudo-abnormal frame, the maximum constraint G_r is used to extend the distance between the pseudo-abnormal frame and the real frame. The inter-frame distance of the pixels between the reconstructed pseudo-abnormal frame and the real frame is increased by the strength constraints L_{sc}, and the difference between the distance between the adjacent pixels in the reconstructed pseudo-abnormal frame and the real frame is increased by the gradient constraints L_{gc}. At the same time, optical flow constraints (L_{opt}) are added to constrain it at the motion level. The maximum constraint G_{max} can be obtained by Formula (14):

$$G_{max} = max(\lambda_{gc}L_{gc} + \lambda_{sc}L_{sc} + \lambda_{opt}L_{opt}) \tag{14}$$

where λ_{gc}, λ_{sc}, and λ_{opt} are the weights of gradient constraints L_{gc}, strength constraints L_{sc}, and optical flow constraints L_{opt}, respectively.

In terms of real frames, the minimum constraint G_{min} is used to narrow the distance between the reconstructed real frames and the real frames. At the apparent level, the inter-frame distance and the intra-frame distance corresponding to the reconstructed real frame and the real frame are shortened, and at the motion level, the optical flow similarity between the reconstructed real frame and the real frame, respectively, and the real frame at the previous moment is improved. The minimum constraint G_{min} is shown in Formula (15):

$$G_{min} = min(\lambda_{gc}L_{gc} + \lambda_{sc}L_{sc} + \lambda_{opt}L_{opt}) \tag{15}$$

Note that the maximum and minimum constraints function can well reduce the problem that the generation ability of the generative confrontation network is too strong, which leads to the perfect reconstruction of abnormal frames. The maximum constraints function is mainly used to increase the distance between the pseudo anomaly frame and the reconstructed pseudo anomaly frame so that the reconstruction generator can recognize the anomaly in advance. At the same time, it makes the reconstruction generator unable to reconstruct similar frames when reconstructing abnormal frames. The minimum constraints function is mainly used to reduce the distance between the original frame and the reconstructed frame so that the reconstruction generator can well learn the normal frame distribution and reconstruct the normal frame.

3.4. Abnormal Detection

DGGAN adds pseudo-anomaly frame generation based on the reconstructed anomaly detection method. Specifically, the pseudo-abnormal frame x' and the original training frame x are used to train the reconstruction generator, so that the generator can recognize the abnormal distribution in advance and improve the perception ability of abnormal. In the test phase, the test frame x'' is input into the trained reconstruction generator to generate a reconstructed test frame. Comparing the reconstructed frame with the real frame, the calculation process of the reconstruction error $S(x'')$ [27] is shown in Formula (16):

$$S(x'') = \lambda L_{app} + (1-\lambda)L_{opt} \tag{16}$$

where λ is a weight parameter.

After obtaining the reconstruction error, we normalize it to obtain the reconstruction score. The calculation process is as in Formula (17):

$$Score(x') = \frac{S(x'') - min(S(x''))}{max(S(x'')) - min(S(x''))} \tag{17}$$

If the reconstruction score of the t-th frame is less than the threshold, it is determined that the frame is a normal frame; otherwise, it is determined that the video frame at this moment contains an abnormal event.

4. Experiments and Results

4.1. Datasets and Train Details

To verify the effectiveness and accuracy of DGGAN, we conduct exhaustive experiments on two datasets, i.e., **UCSD Ped1 and Ped2** and **CUHK Avenue**.

UCSD Ped1 and Ped2 [28]. The scene of the UCSD dataset is a sidewalk, and abnormal events in the scene include bicycles and wheelchairs on the sidewalk, pedestrians running, skateboards, and cars. **Ped1** is the scene of pedestrians moving away and approaching the camera with a resolution of 238 × 158. **Ped2** is a scene in which pedestrians move parallel to the camera. The abnormal events in the scene include bicycles and wheelchairs on the sidewalk, pedestrians running, skateboards and cars, etc. The resolution is 360 × 240.

CUHK Avenue [29]. The scene of the CUHK **Avenue** dataset is Campus Avenue, which contains abnormal events: pedestrians running, walking in the wrong direction, trucks, bicycles, suspicious objects, etc. The resolution is 640 × 360.

All experiments are conducted on the PyTorch deep learning framework with a single RTX 3080Ti (12GB GPU Memory). In both datasets, the training set contains only normal events, while the testing set contains both normal and abnormal events.

The experimental results were evaluated using the area under the curve (AUC) criteria. AUC is an indicator used to evaluate the performance of a binary classifier. It represents the area under the line graph (ROC curve) between the probability of a classifier correctly classifying a positive sample and the probability of misclassifying a negative sample at a certain probability threshold. The larger the AUC value, the better the performance of the classifier, and a value of 1 is a perfect classification. In the video anomaly detection algorithm, the higher the AUC value, the more accurate the algorithm identified the anomaly. Note that the false positive rate (FPR) refers to the ratio of the number of videos that are misjudged as abnormal to the total number of non-abnormal videos among all non-abnormal videos, and the true positive rate (TPR) refers to the ratio of all abnormal videos. The ratio of the number of correctly detected videos to the total number of abnormal videos. The ROC curve is a two-dimensional graph consisting of the TPR and FPR of the classifier at different thresholds.

4.2. Ablation Studies

To verify the effectiveness of the noise generator modules and the second-order channel attention module in DGGAN, we confirm through ablation experiments that these two modules are beneficial to improve the detection accuracy of abnormal video events.

As shown in Table 1, the AUC results of the anomaly detection model with the noise generator on all of these datasets are higher than the anomaly detection model with only the reconstruction generator. Moreover, the anomaly detection model with a noise generator, including SOCA, is higher than the anomaly detection model with the noise generator only.

More specifically, the model without the addition of the three modules is taken as the baseline. Only adding the NGA module will improve the three datasets (i.e., Ped1, Ped2, Avenue, the same as below) by +1.9%, +2.4%, and +0.4%, respectively. When only adding the NGB module, the model has been improved by +0.6%, +1.2%, and +2.2%, respectively. When adding both NGA and NGB, the AUC performance of the model improves by +2.6%, +3.3%, and +1.6%. Note that the AUC performance on the Avenue dataset at this time is not as good as adding only the NGB module, but when adding three modules at the same

time, the performance of the model has been greatly improved (+3.4%, +4.4% and +2.5%). We infer that this is because the combination of NGA and NGB is not completely suitable for all situations, but the combination of these two noise generators and the SOCA module can exert the best effect.

Table 1. Comparison of AUC with noise generator and SOCA. NGA and NGB represent the noise generator modules A and B, respectively. ✔ and ✗ represent our backbone network with or without specific modules, respectively.

Modules			AUC(%)		
NGA	**NGB**	**SOCA**	**PED1**	**PED2**	**Avenue**
✗	✗	✗	82.3	93.5	83.7
✔	✗	✗	84.2	95.9	84.1
✗	✔	✗	82.9	94.7	85.9
✔	✔	✗	84.9	96.8	85.3
✔	✔	✔	**85.7**	**97.9**	**86.2**

The experimental results confirm that compared with the video anomaly detection model trained on normal video frames, the model trained by adding pseudo video frames can detect abnormal video events more accurately.

4.3. Comparison with the State-of-the-Art

As shown in Table 2, we compare DGGAN with SOTA video anomaly detection methods.

Table 2. Comparison of AUC with the SOTA methods.

Methods	AUC(%)		
	PED1	**PED2**	**Avenue**
FastAno [9]	-	96.3	85.3
ASTNet [22]	-	97.4	86.7
R-AVE [30]	75.0	91.0	79.6
Attention Prediction [31]	83.9	96.0	86.0
SSMTL [3]	-	97.5	91.5
Context-related Prediction [15]	-	96.3	87.1
FFP [5]	-	95.4	85.1
DDGAN [16]	82.8	96.7	85.8
DGGAN(ours)	**85.7**	**97.9**	**86.2**

Experiments on the Ped1 dataset show that our method has a certain AUC improvement in detecting anomalies. Concretely, compared with R-VAE, DDGAN, and Attention Prediction, the AUC performance of DGGAN has increased by +10.7%, +2.9%, and +1.8%, respectively. On the Ped2 dataset, although the two SOTA methods, i.e., ASTNet and SSMTL, reach the AUC of 97%, there are still some differences compared with our method. The detection AUC obtained by other methods is significantly lower than our method. This confirms that our method still has some advantages over these two methods in detecting anomalies in Ped2.

On the Avenue dataset, the AUC obtained by ASTNet is only 0.5% higher than that obtained by our method. The detection AUC of SSMTL is significantly higher than our method. We speculate that this is due to the self-supervised learning and teacher–student model used in this method, but this method cannot make the model have a certain awareness of abnormalities in advance during the training phase. The detection of AUCs of other SOTA methods are significantly lower than our method.

It can be seen from the experimental results that since the reconstruction generator uses pseudo video frames for training during training, it has a certain awareness of anomalies in advance. Therefore, when the model discriminates video frames, it reduces the probability

of abnormal frames being reconstructed normally and improves the detection accuracy of the overall method for abnormal events.

5. Conclusions

This paper proposes a new approach to video anomaly detection using the DGGAN (dual-generator generative adversarial network) framework. The method includes a reconstruction generator and a noise generator, each with a unique role. The noise generator generates pseudo-abnormal frames from normal training frames, which in turn trains the reconstruction generator to recognize both normal and abnormal patterns. The DGGAN architecture is further enhanced by introducing a second-order channel attention module, which strengthens the interdependence and correlation of information in the feature maps. Through extensive experiments on two publicly available datasets, CUHK Avenue and UCSD Ped1 and Ped2, the proposed DGGAN method shows a higher detection accuracy compared to other existing methods.

Given that some events are considered abnormal in the video clip, a single frame input may be reconstructed as a normal event. Therefore, when an abnormal event that requires multiple frames to be judged occurs, a single frame and a video package are used for training to improve detection accuracy.

Author Contributions: Conceptualization, X.Q. and Z.H.; methodology, X.Q. and G.J.; software, X.Q.; validation, X.Q. and Z.H.; formal analysis, Z.H.; investigation, X.Q. and Z.H.; resources, G.J.; data curation, X.Q. and Z.H.; writing—original draft preparation, X.Q.; writing—review and editing, Z.H. and G.J.; visualization, X.Q.; supervision, G.J.; project administration, G.J.; funding acquisition, G.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China under grant no. 41971343.

Institutional Review Board Statement: This work does not require ethics approval.

Informed Consent Statement: The research does not involve humans.

Data Availability Statement: The relevant datasets are publicly available for download.

Acknowledgments: The authors would like to thank the editor and reviewers for their constructive comments.

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

References

1. Lu, B.; Xu, D.; Huang, B. Deep-learning-based anomaly detection for lace defect inspection employing videos in production line. *Adv. Eng. Inform.* **2022**, *51*, 101471. [CrossRef]
2. Sun, H.; Chen, M.; Weng, J.; Liu, Z.; Geng, G. Anomaly detection for In-Vehicle network using CNN-LSTM with attention mechanism. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10880–10893. [CrossRef]
3. Georgescu, M.I.; Barbalau, A.; Ionescu, R.T.; Khan, F.S.; Popescu, M.; Shah, M. Anomaly detection in video via self-supervised and multi-task learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12742–12752.
4. Zhong, J.X.; Li, N.; Kong, W.; Liu, S.; Li, T.H.; Li, G. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1237–1246.
5. 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.
6. Rodrigues, R.; Bhargava, N.; Velmurugan, R.; Chaudhuri, S. Multi-timescale trajectory prediction for abnormal human activity detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 2626–2634.
7. Li, N.; Chang, F.; Liu, C. Spatial-temporal cascade autoencoder for video anomaly detection in crowded scenes. *IEEE Trans. Multimed.* **2020**, *23*, 203–215. [CrossRef]
8. Georgescu, M.I.; Ionescu, R.T.; Khan, F.S.; Popescu, M.; Shah, M. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 4505–4523. [CrossRef] [PubMed]

9. Park, C.; Cho, M.; Lee, M.; Lee, S. FastAno: Fast anomaly detection via spatio-temporal patch transformation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2022; pp. 2249–2259.
10. Gong, D.; Liu, L.; Le, V.; Saha, B.; Mansour, M.R.; Venkatesh, S.; Hengel, A.V.D. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 1705–1714.
11. Park, H.; Noh, J.; Ham, B. Learning memory-guided normality for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14372–14381.
12. Ionescu, R.T.; Khan, F.S.; Georgescu, M.I.; Shao, L. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7842–7851.
13. Aldausari, N.; Sowmya, A.; Marcus, N.; Mohammadi, G. Video generative adversarial networks: A review. *ACM Comput. Surv. (CSUR)* **2022**, *55*, 1–25. [CrossRef]
14. Vu, T.H.; Boonaert, J.; Ambellouis, S.; Taleb-Ahmed, A. Multi-Channel Generative Framework and Supervised Learning for Anomaly Detection in Surveillance Videos. *Sensors* **2021**, *21*, 3179. [CrossRef] [PubMed]
15. Li, D.; Nie, X.; Li, X.; Zhang, Y.; Yin, Y. Context-related video anomaly detection via generative adversarial network. *Pattern Recognit. Lett.* **2022**, *156*, 183–189. [CrossRef]
16. Xu, J.; Miao, Z.; Xu, W.; Wang, J.; Zhang, Q.; Song, S. Video Anomaly Detection Using Dual Discriminator Based Generative Adversarial Network. In Proceedings of the 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, Pasadena, CA, USA, 13–15 December 2021; pp. 1259–1265.
17. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
18. Liu, Y.; Shao, Z.; Hoffmann, N. Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. *arXiv* **2021**, arXiv:2112.05561.
19. Zhang, S.; Gong, M.; Xie, Y.; Qin, A.; Li, H.; Gao, Y.; Ong, Y.S. Influence-aware Attention Networks for Anomaly Detection in Surveillance Videos. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 5427–5437. [CrossRef]
20. Khorramshahi, P.; Peri, N.; Kumar, A.; Shah, A.; Chellappa, R. Attention Driven Vehicle Re-identification and Unsupervised Anomaly Detection for Traffic Understanding. In Proceedings of the CVPR Workshops, Long Beach, CA, USA, 16–20 June 2019; pp. 239–246.
21. Feng, J.C.; Hong, F.T.; Zheng, W.S. Mist: Multiple instance self-training framework for video anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14009–14018.
22. Le, V.T.; Kim, Y.G. Attention-based residual autoencoder for video anomaly detection. *Appl. Intell.* **2022**, *53*, 3240–3254. [CrossRef]
23. Dai, T.; Cai, J.; Zhang, Y.; Xia, S.T.; Zhang, L. Second-order attention network for single image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 11065–11074.
24. Li, P.; Xie, J.; Wang, Q.; Zuo, W. Is second-order information helpful for large-scale visual recognition? In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2070–2078.
25. Piga, N.A.; Onyshchuk, Y.; Pasquale, G.; Pattacini, U.; Natale, L. ROFT: Real-Time Optical Flow-Aided 6D Object Pose and Velocity Tracking. *IEEE Robot. Autom. Lett.* **2021**, *7*, 159–166. [CrossRef]
26. Müller, R.; Kornblith, S.; Hinton, G.E. When does label smoothing help? *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
27. Akçay, S.; Atapour-Abarghouei, A.; Breckon, T.P. Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. In Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.
28. Mahadevan, V.; Li, W.; Bhalodia, V.; Vasconcelos, N. Anomaly detection in crowded scenes. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1975–1981.
29. Lu, C.; Shi, J.; Jia, J. Abnormal event detection at 150 fps in matlab. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 2720–2727.
30. Yan, S.; Smith, J.S.; Lu, W.; Zhang, B. Abnormal event detection from videos using a two-stream recurrent variational autoencoder. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *12*, 30–42. [CrossRef]
31. Zhou, J.T.; Zhang, L.; Fang, Z.; Du, J.; Peng, X.; Xiao, Y. Attention-driven loss for anomaly detection in video surveillance. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 4639–4647. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.