IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 January 2024, accepted 26 February 2024, date of publication 7 March 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3374383

RESEARCH ARTICLE

Making Anomalies More Anomalous: Video Anomaly Detection Using a Novel Generator and Destroyer

SEUNGKYUN HONG, SUNGHYUN AHN, YOUNGWAN JO, AND SANGHYUN PARK, (Member, IEEE)
Department of Computer Science, Yonsei University, Seodaemun-gu, Seoul 03722, South Korea

Corresponding author: Sanghyun Park (sanghyun@yonsei.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by Korean Government through the Ministry of Science and ICT (MSIT) [(SW Star Laboratory) Research and Development of the High Performance In-Memory Distributed DBMS Based on Flash Memory Storage in the IoT Environment] under Grant IITP-2017-0-00477, and in part by the Artificial Intelligence Graduate School Program (Yonsei University) under Grant 2020-0-01361.

ABSTRACT We propose a novel approach for video anomaly detection. Existing video anomaly detection methods train only on normal frames, with the expectation that the quality of the abnormal frames will decrease, and utilize the reconstruction error with the ground truth to detect anomalies. However, a challenge exists owing to the powerful generalization capability of deep neural networks, as they tend to proficiently generate abnormal frames. To address this issue, we introduce a novel method to make anomalies more anomalous by destroying abnormal areas in abnormal frames. Accordingly, we propose the frame-to-label and motion (F2LM) generator and Destroyer. The F2LM generator predicts a future frame by utilizing the label and motion information of the input frames, thereby degrading the quality of abnormal regions. The Destroyer destroys abnormal regions by transforming low-quality areas into zero vectors. Both models were trained individually, and during testing, the F2LM generator degraded the quality of abnormal regions, and the Destroyer subsequently destroyed these areas. Our proposed video anomaly detection method demonstrated superior performance compared to state-of-the-art models with three benchmark datasets (UCSD Ped2, CUHK Avenue, Shanghai Tech.). Our code and models are available online at https://github.com/SkiddieAhn/Paper-Making-Anomalies-More-Anomalous.

INDEX TERMS Deep learning, future frame prediction, video anomaly detection, video surveillance.

I. INTRODUCTION

Intelligent video surveillance systems are crucial infrastructure for preventing accidents and ensuring swift response after incidents, as they are able to analyze video information to automatically detect abnormal behavior. Therefore, video anomaly detection is crucial for establishing a safety network for society.

In the field of video anomaly detection, unsupervised learning methods are commonly employed because abnormal situations do not occur frequently. Two predominant unsupervised learning methods are utilized for video anomaly detection: reconstruction-based [1], [2], [3], [4], [5], [6], [7], [8], [9] and prediction-based methods [5], [10], [11], [12], [13], [14], [15], [16], [17]. During training, both methods learn the characteristics of normal frames. In testing, when abnormal frames are encountered, they are reconstructed or predicted to be normal. The discrepancy between the reconstructed or predicted frames and the actual abnormal frames is then used to identify anomalies. In general, reconstruction-based and prediction-based methods use encoder-decoder based generators to reconstruct or predict abnormal frames. However, according to recent studies [18], [19], [20], [21], because of the strong generalization capacity of deep neural networks, contrary to expectations, these methods can effectively generate anomalous frames, as shown in Fig. 1.

The associate editor coordinating the review of this manuscript and approving it for publication was Chaker Larabi.

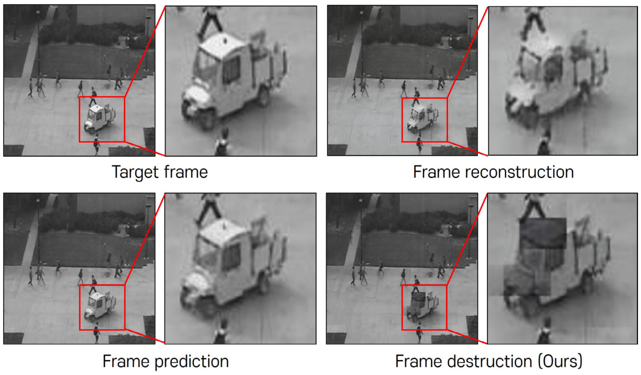


FIGURE 1. **Image quality comparison of the anomalous regions in a generated frame for reconstruction-based model, prediction-based model, and our proposed model.**

These studies have indicated that even abnormal situations, such as objects or actions that were not visible during training, can be generated. To solve this problem, previous studies have used pseudo-anomaly images [18] or self-supervised methods for learning abnormal situations [20], [21]. Recently, a method for storing latent vectors of normal frames utilizing a memory mechanism has also been proposed [5], [6]. This approach utilizes the feature representations of normal frames stored in memory to generate frames, thereby reducing the ability to generate abnormal frames. However, these methods heavily depend on memory size, and smaller memory sizes may not adequately store the diverse feature representations of normal frames. Consequently, this limitation may also affect the capability to generate normal data [9], [22].

To address these issues, we propose a frame-to-label and motion (F2LM) generator that predicts a future frame with lower quality in abnormal regions, and a Destroyer that transforms low-quality areas into zero vectors, thus destroying the abnormal regions.

The F2LM generator is a prediction-based network that takes four consecutive video frames as input to predict the future frame. It is trained in an unsupervised manner. The goal of the F2LM generator is to predict low-quality future frames in abnormal video sequences. To achieve this, we propose a feature transform convolutional block (FTC) and use triplet loss [23] for this training. First, we obtain video sequences for both label and motion from the input video sequence using DeepLabv3 [24] and FlowNet2 [25], respectively. Then, we extract features for frame, label, and motion using individual encoders. The FTC block is trained to transform the frame feature into label and motion features. Triplet loss encourages the FTC block to transform the frame feature to be similar to either the label or motion feature from the encoders while pushing the transformed features away from the frame feature. Thus, when an abnormal video sequence is input, the FTC block struggles to transform the label or motion feature, which leads to low-quality future frame prediction. Therefore, the F2LM generator effectively predicts a future frame for a normal video sequence but struggles to predict a future frame for an abnormal video sequence.

The Destroyer is a network that takes the future frame generated by the F2LM generator as the input and aims to destroy low-quality abnormal areas. During training, because only normal data are available, it uses a self-supervised approach by adding noise to the future frame to create arbitrary abnormal areas for training. The goal of the Destroyer is to identify low-quality areas as abnormal regions and transform them into zero vectors to make the future frame appear even more anomalous. To create arbitrary abnormal regions, we divide the future frame into non-overlapping patches and add random noise to some patches. The Destroyer is trained to transform noisy patches into zero vectors and to reconstruct patches with no added noise. Noise is not added during testing, and the Destroyer destroys the low-quality areas in the future frame generated by the F2LM generator. Therefore, the Destroyer increases the quality difference between normal and abnormal frames by destroying abnormal regions, thereby improving video anomaly detection performance.

We utilize the F2LM generator and Destroyer together to destroy abnormal areas in video frames, thereby increasing the difference in anomaly scores between normal and abnormal frames, as shown in Frame destruction (Ours) in Fig. 1. Our proposed model achieves superior results compared to state-of-the-art models on video anomaly detection benchmarks. The contributions of this study are as follows.

- We propose the novel F2LM generator, which utilizes a training strategy to transform the feature of an input video sequence into label and motion features and focuses on learning transformations that do not perform effectively with abnormal video sequences. Through this process, when an abnormal video sequence is provided, the model predicts a low-quality future frame.
- We propose the novel Destroyer, which identifies low-quality areas as abnormal regions and turns them into zero vectors, thereby destroying them. The Destroyer significantly enhances the anomaly score difference between a normal and abnormal frame, thus improving video anomaly detection performance.
- We evaluate the F2LM generator and Destroyer using three public benchmarks. The results demonstrate that our approach outperforms other state-of-the-art methods.

II. RELATED WORKS

A. RECONSTRUCTION-BASED METHODS

Reconstruction-based methods train a model to minimize the reconstruction error of a normal video sequence. The expectation is that once trained in this manner, the model will struggle to effectively reconstruct abnormal events during the testing phase. Chong and Tay [3] proposed a method in which normal frames are input into a convolutional long short-term memory (ConvLSTM) [2] to extract features

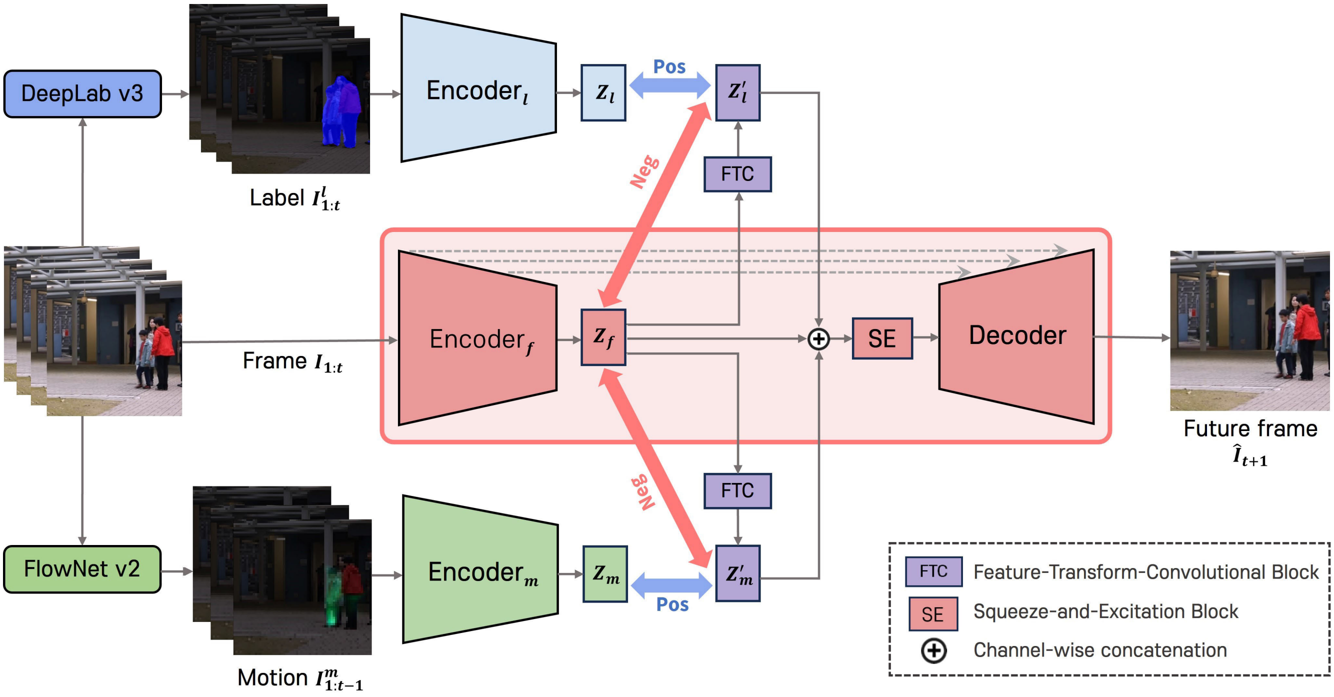


FIGURE 2. **The architecture of the frame-to-label and motion (F2LM) generator. The F2LM generator includes a module to transform a frame feature into label and motion features in bottleneck areas. If a frame with abnormal objects or behavior is input, the transformation from frame features into other features becomes challenging, limiting the prediction of future frames.**

with spatio-temporal information. These features are then reconstructed using a decoder. Nguyen and Meunier [8] propose an approach that uses one encoder and two decoders. The first decoder is trained to reconstruct the input frames, whereas the second is trained to predict the optical flow representing the movement between time steps t and $t+1$. This structure aims to maximize both the reconstruction and prediction errors for abnormal data. Further Zhong et al. [9] attempted video anomaly detection by connecting a network that reconstructs video frames with one that predicts the magnitude (size) and orientation (direction) of the optical flow. Through this structure, when abnormal data are input, the goal is to make reconstructed frames of low-quality increase the optical flow prediction error.

B. PREDICTION-BASED METHODS

Prediction-based methods involve training a model to take input from a normal video sequence and predict a future frame. The expectation is that the model, once trained in this manner, will struggle to predict an abnormal future frame during the testing phase. Liu et al. [10] were the first to propose a prediction-based method using a generative adversarial network (GAN) [26] structure. They employed the U-Net [27] as a generator and utilized the PatchGAN classifier from pix2pix [28] as a discriminator to predict a more realistic future frame. Additionally, they trained the model to approximate real optical flow by leveraging a pre-trained FlowNet2 [25]. Ye et al. [11] proposed a model called AnoPCN, which consists of a predictive coding module (PCM) and an error refinement module (ERM). The PCM predicts future frames by inputting reconstructed previous frames and reconstruction errors (RGB difference) into a ConvLSTM. The ERM module improves the quality of the predicted future frames by adding prediction errors to the future frames predicted by the PCM. This structure explicitly utilizes reconstruction errors as motion information, reducing prediction errors for normal future frames. Moreover Yuan et al. [12] proposed a video anomaly detection model named TransAnomaly based on U-Net and the video vision transformer (ViViT) [29]. The convolutional neural network (CNN) features extracted by the encoder part of U-Net are encoded by a modified ViViT. As a result, the encoded features provide better spatio-temporal information. When reconstructed using the decoder, the model reduced the prediction errors for a normal future frame compared to existing methods.

Thus, deep learning-based reconstruction and prediction methods for video anomaly detection have demonstrated promising performance through various approaches. However, according to several studies [18], [19], [20], [21], deep neural networks utilizing CNN architectures in reconstruction-based and prediction-based methods are adept at generating abnormal frames owing to their powerful generalization capabilities. Therefore, these two methods cannot ensure discrimination between anomaly scores for normal and abnormal frames.

C. MEMORY-BASED METHODS

Memory-based methods have been introduced to address the challenge of effectively generating abnormal frames.

MemAE [6] aggregates the most similar features from the information stored in memory, which correspond to the features extracted by the encoder for the input image, and sends it to the decoder. MNAD [5] introduces multiple prototypes to capture various patterns in normal videos. The CNN is trained using feature separateness loss, which encourages similar features to form clusters. These memory-based methods employ an approach in which, for abnormal frames, similar information is not stored in memory, resulting in a higher reconstruction error for the decoder's output. However, these memory-based methods may encounter limitations in fully learning representations for normal frames depending on the size of the memory. This constraint could restrict the generative capacity for normal frames.

III. METHOD

A. F2LM GENERATOR

The F2LM generator comprises four main components: an encoder, a feature transform convolutional (FTC) block, a squeeze and excitation (SE) block [30], and a decoder. The overall architecture follows the structure of U-Net [27], with the bottleneck composed of the FTC and SE blocks. The detailed structures of the encoder and decoder are presented in Section IV-B1. Fig. 2 illustrates the overall structure of the F2LM generator. Detailed explanations of each component are as follows.

1) ENCODER

The F2LM generator comprises three encoders, E_f, E_l, E_m, each responsible for extracting frame, label, and motion features. Given a sequence of continuous video frames $I_1, I_2, \ldots, I_t \in \mathbb{R}^{H\times W\times 3}$, DeepLabv3 and FlowNet2 are utilized to generate label sequence $I_1^l, I_2^l, \ldots, I_t^l \in \mathbb{R}^{H\times W\times 21}$ and motion sequence $I_1^m, I_2^m, \ldots, I_{t-1}^m \in \mathbb{R}^{H\times W\times 2}$. These generated sequences pass through their respective encoders, producing the frame feature $Z_f = E_f(I_{1:t})$, label feature $Z_l = E_l(I_{1:t}^l)$, and motion feature $Z_m = E_m(I_{1:t-1}^m)$. DeepLabv3 is a model used in semantic segmentation tasks and can effectively label images on a pixel basis. FlowNet2 is a model used in optical flow estimation tasks and is effective in identifying visual movement by quantifying the pixel movement between video frames. Because both models also show good generalization performance with benchmark datasets for video anomaly detection, we extract label and motion features to utilize information regarding the object class of the video sequence and the motion or movement of each object.

2) FTC BLOCK

The FTC block, which is composed of 2D convolutional layers, transforms Z_f into the frame-to-label feature Z_l' and frame-to-motion feature Z_m'. We use triplet loss to ensure that Z_l' and Z_m' generated from Z_f are similar to Z_l and Z_m, respectively. Triplet loss aims to minimize the distance between an anchor and a positive feature while maximizing the distance between the anchor and a negative feature.

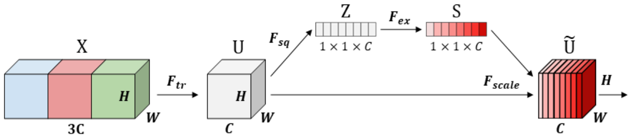


FIGURE 3. The architecture of the squeeze and excitation block.

For label and motion, we use Z_l' and Z_m' as anchors, Z_l and Z_m as positive features, and Z_f as a negative feature. The formulation for the triplet loss is as follows:

$$L_{triplet}^{l}(Z_f, Z_l, Z_l') = \max\{d(Z_l', Z_l) - d(Z_l', Z_f) + \alpha, 0\}, \tag{1}$$

$$L_{triplet}^{m}(Z_f, Z_m, Z_m') = \max\{d(Z_m', Z_m) - d(Z_m', Z_f) + \alpha, 0\}, \tag{2}$$

$$\begin{aligned} L_{triplet}(Z_f, Z_l, Z_m, Z_l', Z_m') = {} & L_{triplet}^{l}(Z_f, Z_l, Z_l') \\ & + L_{triplet}^{m}(Z_f, Z_m, Z_m'), \end{aligned} \tag{3}$$

where $d(x, y)$ represents the Euclidean distance between vectors x and y. $L_{triplet}^{l}$ and $L_{triplet}^{m}$ denote the label and motion triplet losses, respectively. α is a hyperparameter to increase the distance between the anchor and the negative feature.

The structure using the FTC block and triplet loss is designed to penalize abnormal video sequences. An FTC block trained exclusively on a normal video sequence will transform Z_f into Z_l' and Z_m' with minimal differences between Z_l and Z_m for a normal video sequence. However, for abnormal video sequences, the differences between Z_l, Z_m and Z_l', Z_m' increase. Therefore, using Z_f, Z_l', and Z_m' generated from abnormal video sequences to predict the future frame results in poor generation, increases the anomaly score and contributes to performance improvement.

3) SE BLOCK

After channel-wise concatenating Z_f with Z_l' and Z_m' generated by the FTC block, we use this concatenated feature as the input for the SE block. The SE block is identical to the structure proposed by Hu et al. [30], as shown in Fig. 3. The SE block is an attention module that focuses on important channels within the channel information. Formally, the input feature $\mathrm{X} \in \mathbb{R}^{H\times W\times 3C}$, undergoes a 3×3 convolution to produce $\mathrm{U} \in \mathbb{R}^{H\times W\times C}$ and global pooling is performed on each channel, resulting in the vector $\mathrm{Z} \in \mathbb{R}^{1\times 1\times C}$. Subsequently, the vector of scale factor $\mathrm{S} \in \mathbb{R}^{1\times 1\times C}$ is calculated as follows:

$$\mathrm{S} = \sigma(\delta(\mathrm{Z} \cdot W_1) \cdot W_2), \tag{4}$$

where σ is the sigmoid activation, δ is the ReLU activation, $W_1 \in \mathbb{R}^{C\times\frac{C}{r}}$ and $W_2 \in \mathbb{R}^{\frac{C}{r}\times C}$ represent the weight matrices of two consecutive fully connected layers, and r is the reduction ratio.

The last step is the multiplication of S and U on each channel, producing the final tensor $\tilde{\mathrm{U}} \in \mathbb{R}^{H\times W\times C}$ containing recalibrated features maps [31]. The SE block has the advantage of adaptively recalibrating features with few

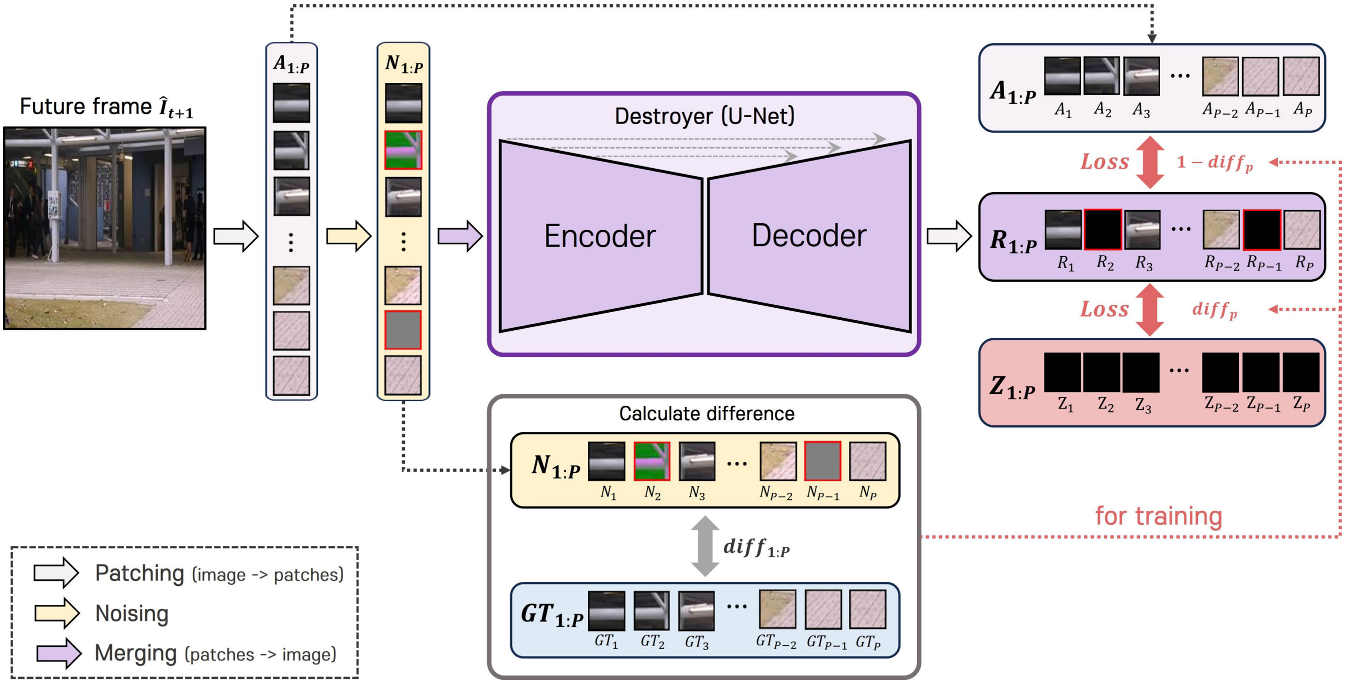

Future frame $\hat{I}_{t+1}$
$A_{1:P}$
$N_{1:P}$
Destroyer (U-Net)
Encoder
Decoder
$A_{1:P}$
A_1 A_2 A_3 A_{P-2} A_{P-1} A_P
Loss
$1-diff_p$
$R_{1:P}$
R_1 R_2 R_3 R_{P-2} R_{P-1} R_P
Loss
$diff_p$
$Z_{1:P}$
Z_1 Z_2 Z_3 Z_{P-2} Z_{P-1} Z_P
Calculate difference
$N_{1:P}$
N_1 N_2 N_3 N_{P-2} N_{P-1} N_P
$diff_{1:P}$
$GT_{1:P}$
GT_1 GT_2 GT_3 GT_{P-2} GT_{P-1} GT_P
for training
Patching (image -> patches)
Noising
Merging (patches -> image)

FIGURE 4. The architecture of the Destroyer. The Destroyer takes a generated future frame as the input and detects low-quality regions and destroys them. This enhances the abnormality in the output.

parameters, thereby allowing crucial channels from the three features to pass to the decoder. This advantage enhances the generation capability of future frames, ultimately improving anomaly detection performance by reducing false positives. Table 4 shows the related experiments. Finally, the future frame $\hat{I}_{t+1} \in \mathbb{R}^{H\times W\times 3}$ is generated by the decoder.

B. DESTROYER

The Destroyer destroys low-quality regions in the future frame predicted by the F2LM generator, considering them anomalous areas. We employed a method for destroying abnormal regions by transforming them into zero vectors. During training, because the training data consists only of normal frames, the F2LM generator is unable to generate low-quality regions. Therefore, we add noise to the output of the F2LM generator, generating arbitrary abnormal frames to be input into the Destroyer. During testing, when the F2LM generator predicts abnormal regions as low-quality noise, the Destroyer makes these areas even more anomalous. Fig. 4 illustrates the overall structure of the proposed Destroyer.

1) PATCHING AND NOISING

Given the future $\hat{I}_{t+1} \in \mathbb{R}^{H\times W\times C}$ predicted by the F2LM generator, it is divided into P patches $A_1, A_2, \ldots, A_p \in \mathbb{R}^{H'\times W'\times C}$, where $H' = \frac{H}{\sqrt{P}}$ and $W' = \frac{W}{\sqrt{P}}$. Among these P patches, a random number of patches between 5% and 50% have noise injected. Noise injection is achieved through dropout, and the dropout rate is randomly set between 5% and 50%. P patches with injected noise $N_1, N_2, \ldots, N_p \in \mathbb{R}^{H'\times W'\times C}$ are merged and serve as input for the encoder of the Destroyer. The locations and number of patches to which noise is applied are randomly set to handle abnormal regions of various sizes and positions. Additionally, to account for the diversity in the generator's prediction of abnormal regions as noise, the dropout rate is also randomly set during training.

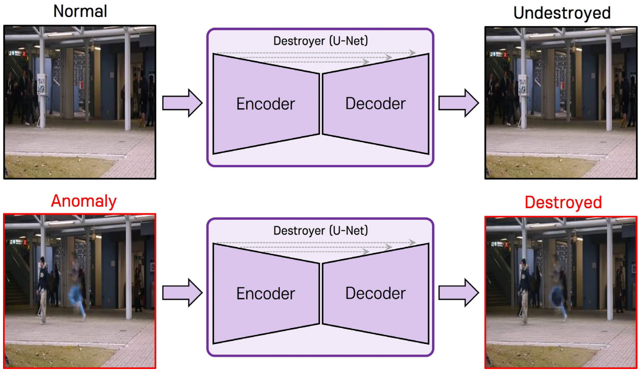


FIGURE 5. Comparison of the future frame with the application of the Destroyer for normal and anomalous video frames.

2) DESTROYER LOSS

We propose a Destroyer loss function for training the Destroyer. First, the ground truth frame is divided into P patches, denoted by $GT_{1:P}$; each are the same size as the patches in the frame with injected noise $N_{1:P}$. The difference between each corresponding patch of $N_{1:P}$ and $GT_{1:P}$, denoted by $diff_{1:P}$, are then calculated. $diff_p$ for the p^{th} patch is calculated as follows:

$$diff_p = \mathrm{MIN}(\lambda(1 - \mathrm{SSIM}(N_p, GT_p)), 1), \tag{5}$$

where λ is a hyperparameter to set the learning direction of the Destroyer.

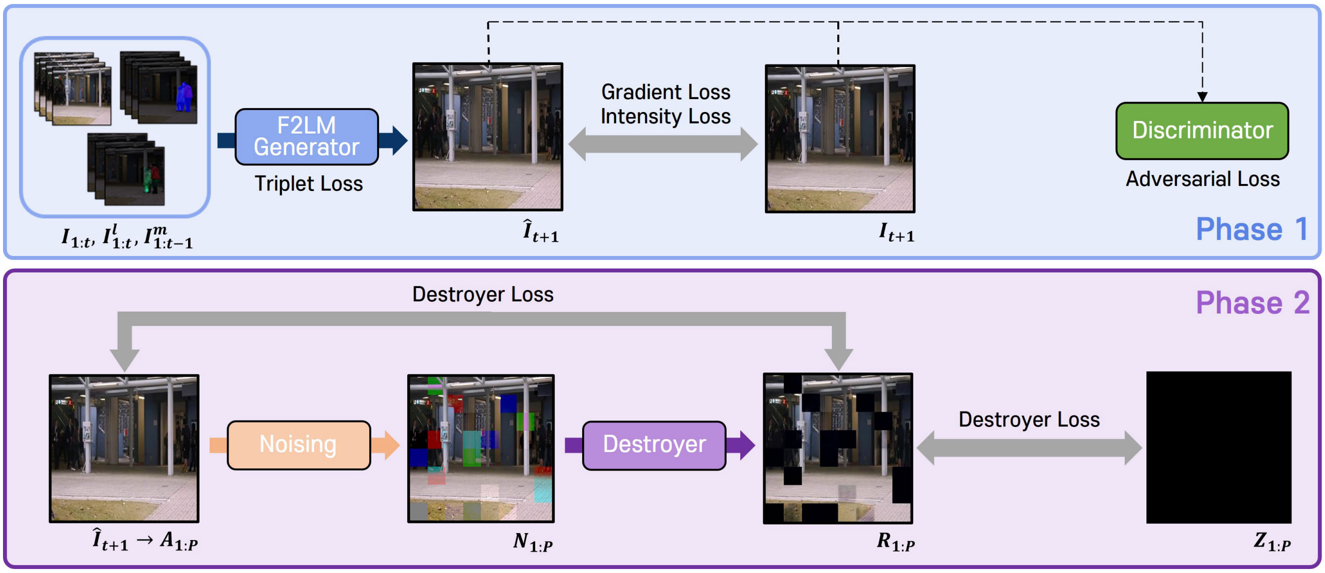


FIGURE 6. Model Training Process. Phase 1 consists of unsupervised learning for the F2LM generator, while Phase 2 consists of self-supervised learning for the Destroyer.

$diff_p$ takes values between 0 and 1 and is used as a measure to distinguish between normal and abnormal for each patch. We use the structural similarity index measure (SSIM) [32] to determine the image quality difference between the injected noise patches and the ground truth. Therefore, a smaller $diff_p$ value indicates a normal patch, while a larger $diff_p$ value indicates an abnormal patch. Subsequently, the video frame R generated by the Destroyer is divided into P patches $R_{1:P}$. For each patch, if the $diff_p$ value is closer to 1, it guides the Destroyer to transform R_p into a zero vector (Z_p). Otherwise, if it is closer to zero, it guides the Destroyer to reconstruct R_p for the future frame patch unit generated by the F2LM generator (A_p). The Destroyer loss, $L_{Destroyer}$, is calculated as

$$L_{Destroyer}(diff, R, Z, A) = \sum_{p=1}^{P}(diff_p \cdot \|Z_p - R_p\|_2^2 + (1 - diff_p) \cdot \|A_p - R_p\|_2^2). \quad (6)$$

Therefore, during testing, the Destroyer, trained through the Destroyer loss, further destroys abnormal areas with degraded quality in the future frame predicted by the F2LM generator, enhancing anomaly detection performance.

3) TEST

During testing, the input frames are not subjected to the patching and noising processes as in training. Fig. 5 illustrates the testing process of the Destroyer. While it reconstructs normal frames as they are, for abnormal frames, it further enhances their abnormal characteristics.

C. TRAINING METHOD

Our proposed framework follows a two-stage process in which the F2LM generator is first trained, followed by the Destroyer training. Fig. 6 illustrates the model training process, with Phase 1 and 2 dedicated the training of the F2LM generator and Destroyer, respectively. The F2LM generator employs four objective functions: intensity loss, gradient loss, adversarial loss, and triplet loss. Triplet loss is explained in Section III-A. To ensure the predicted future frame $\hat{I}$ closely resembles the target future frame I, intensity and gradient losses are utilized. The intensity loss ensures the similarity of all the pixels in the RGB space, whereas the gradient loss reduces the difference between the predicted and ground truth images' surrounding pixels to make the two images more alike. The intensity and gradient losses are defined as (7) and (8), respectively:

$$L_{int}(\hat{I}, I) = \|\hat{I} - I\|_2^2, \quad (7)$$

$$L_{gd}(\hat{I}, I) = \sum_{i,j} \| |\hat{I}_{i,j} - \hat{I}_{i-1,j}| - |I_{i,j} - I_{i-1,j}| \|_1 + \| |\hat{I}_{i,j} - \hat{I}_{i,j-1}| - |I_{i,j} - I_{i,j-1}| \|_1, \quad (8)$$

where i and j represent spatial indices of a video frame.

In addition, to enhance the F2LM generator's ability to predict a more realistic future frame, we employ the least squares generative adversarial network (GAN) structure proposed in [33]. The least squares GAN consists of a generator (G) and a discriminator (D). D is trained to effectively classify the ground truth and predicted images, whereas G is trained to generate predicted images that resemble the ground truth, making D's classification challenging. We use the F2LM generator as G and the PatchGAN classifier from pix2pix as D. The adversarial loss for training discriminator (D) and generator (G) is given by (9) and (10), respectively:

$$L_{adv}^{D}(\hat{I}, I) = \sum_{i,j} \frac{1}{2} L_{MSE}(D(I)_{i,j}, 1) + \sum_{i,j} \frac{1}{2} L_{MSE}(D(\hat{I})_{i,j}, 0), \quad (9)$$

$$L_{adv}^{G}(\hat{I}) = \sum_{i,j} \frac{1}{2} L_{MSE}(D(\hat{I})_{i,j}, 1), \quad (10)$$

where *i* and *j* represent spatial indices of a video frame. MSE denotes mean square error.

D is trained to output 1 for the ground truth frame and 0 for the predicted frame depending on the input. G is trained to have a predicted frame output of 1 when fed into D. The final loss function for training the F2LM generator is

$$\begin{aligned} L_{Generator} &= \delta_{int} L_{int}(\hat{I}, I) + \delta_{gd} L_{gd}(\hat{I}, I) \\ &\quad + \delta_{adv} L_{adv}^{G}(\hat{I}) + \delta_{tri} L_{triplet}(Z_f, Z_l, Z_m, Z_l', Z_m'), \end{aligned} \tag{11}$$

where δ_{int}, δ_{gd}, δ_{adv}, and δ_{tri} denote hyperparameters that control the influence of each loss.

The Destroyer uses the Destroyer loss for training, as explained in Section III-B.

D. ANOMALY SCORE

The anomaly score is used as a metric to determine normal and abnormal conditions, obtained by calculating the scaled score from the F2LM generator and Destroyer, followed by normalization. The scaled score is given by (12), where *SL* denotes the scaled value of the loss function, calculated as shown in (13) and (14):

$$\begin{aligned} Scaled\ Score &= \gamma_1 \cdot SL_{triplet}^{l} + \gamma_2 \cdot SL_{triplet}^{m} \\ &\quad + \gamma_3 \cdot SL_{MSE}^{Generator} + \gamma_4 \cdot SL_{MSE}^{Destroyer}, \end{aligned} \tag{12}$$

$$SL_{triplet} = \frac{L_{triplet} - \mu(L_{triplet})}{\sigma(L_{triplet})}, \tag{13}$$

$$SL_{MSE} = \frac{L_{MSE} - \mu(L_{MSE})}{\sigma(L_{MSE})}, \tag{14}$$

where the parameters $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are hyperparameters that control the influence of each term and can vary depending on the dataset. μ and σ denote mean and standard deviation, respectively.

The first and second terms of the scaled score are scaled values of the triplet loss for label and motion, respectively. The third and fourth terms are scaled values of the MSE loss between the outputs of the F2LM generator and Destroyer, and the ground truth. This evaluation facilitates the assessment of how well frame-to-label and frame-to-motion are performed at the feature level and how closely the generated frames resemble the ground truth at the frame level. This enables abnormal situations to be detected. Finally, the anomaly score is obtained by min-max normalization of the scaled score and is represented by (15):

$$\begin{aligned} &Anomaly\ Score \\ &\quad = \frac{Scaled\ Score - \mathrm{MIN}(Scaled\ Score)}{\mathrm{MAX}(Scaled\ Score) - \mathrm{MIN}(Scaled\ Score)}. \end{aligned} \tag{15}$$

IV. EXPERIMENTS

A. DATASET

Three benchmark datasets are used to evaluate the proposed method. The training videos for each dataset consist only of normal videos, whereas the testing videos include both normal and abnormal videos.

TABLE 1. Detailed network architecture of the encoder in the F2LM generator. Abbreviations: *k*: kernel, *p*: padding, *s*: stride, *H*: height, *W*: width, *C*: channel.

Layer name	Layer	Input size ($H \times W \times C$)	Output size ($H \times W \times C$)
inconv	$\begin{bmatrix} \text{Conv2D } (k{=}3, p{=}1, s{=}1) \\ \text{BatchNorm2d} \\ \text{ReLU} \end{bmatrix} \times 2$	$256 \times 256 \times C$	$256 \times 256 \times 64$
downconv$_1$	MaxPool2d(k=2, s=2) $\begin{bmatrix} \text{Conv2D } (k{=}3, p{=}1, s{=}1) \\ \text{BatchNorm2d} \\ \text{ReLU} \end{bmatrix} \times 2$	$256 \times 256 \times 64$	$128 \times 128 \times 128$
downconv$_2$	MaxPool2d(k=2, s=2) $\begin{bmatrix} \text{Conv2D } (k{=}3, p{=}1, s{=}1) \\ \text{BatchNorm2d} \\ \text{ReLU} \end{bmatrix} \times 2$	$128 \times 128 \times 128$	$64 \times 64 \times 256$
downconv$_3$	MaxPool2d(k=2, s=2) $\begin{bmatrix} \text{Conv2D } (k{=}3, p{=}1, s{=}1) \\ \text{BatchNorm2d} \\ \text{ReLU} \end{bmatrix} \times 2$	$64 \times 64 \times 256$	$32 \times 32 \times 512$
LN	L_2 normalization	$32 \times 32 \times 512$	$32 \times 32 \times 512$

- The UCSD Ped2 dataset consists of 16 training videos and 12 testing videos. Abnormal situations in this dataset involve videos of pedestrians or vehicles moving on roads [34].
- The CUHK Avenue dataset consists of 16 training videos and 21 testing videos. Abnormal situations in this dataset include videos with behaviors such as throwing objects, loitering, and running. It is important to note that, due to camera perspectives, objects of the same identity may be different sizes [35].
- The Shanghai Tech. dataset comprises 330 training videos and 107 testing videos, representing 13 different scenes. This dataset includes a wide range of abnormal situations, such as the appearance of abnormal objects like bicycles and cars, as well as abnormal behaviors like falling. It is considered a highly challenging dataset [36].

B. IMPLEMENTATION DETAILS

The F2LM generator and discriminator are optimized using the AdamW [37] optimizer, with the learning rates set to 2e-4 and 2e-5, respectively. For the UCSD Ped2 dataset, learning rates of 1e-4 and 1e-5 are used. Additionally, weight decay uses the same value as the learning rate of the F2LM generator, and the iteration and batch size are set to 60,000 and 4, respectively. A single NVIDIA GeForce RTX 3090 graphics card is used in the experiment. The number of training iterations for the Destroyer is set to 15,000, and the remaining training configurations are the same as those for the F2LM generator.

1) NETWORK DESIGN

a: F2LM GENERATOR

The F2LM generator is based on a modified version of U-Net used by Liu et al. [10]. U-Net generates future frames effectively by passing the hierarchical spatio-temporal features extracted from each layer of the encoder to the decoder using skip connections. The encoder and decoder

TABLE 2. Detailed network architecture of the decoder in the F2LM generator. Abbreviations: *k*: kernel, *p*: padding, *s*: stride, *H*: height, *W*: width, *C*: channel.

Layer name	Layer	Input size $(H \times W \times C)$	Output size $(H \times W \times C)$
upconv$_1$	ConvTranspose2d(*k*=2,*s*=2) channel-wise concat [Conv2D (*k*=3, *p*=1,*s*=1) BatchNorm2d ReLU] × 2	$32 \times 32 \times 512$ $64 \times 64 \times 256$	$64 \times 64 \times 256$
upconv$_2$	ConvTranspose2d(*k*=2,*s*=2) channel-wise concat [Conv2D (*k*=3, *p*=1,*s*=1) BatchNorm2d ReLU] × 2	$64 \times 64 \times 256$ $128 \times 128 \times 128$	$128 \times 128 \times 128$
upconv$_3$	ConvTranspose2d(*k*=2,*s*=2) channel-wise concat [Conv2D (*k*=3, *p*=1,*s*=1) BatchNorm2d ReLU] × 2	$128 \times 128 \times 128$ $256 \times 256 \times 64$	$256 \times 256 \times 64$
outconv	Conv2D (*k*=3, *p*=1,*s*=1) tanh	$256 \times 256 \times 64$	$256 \times 256 \times 3$

TABLE 3. The detailed network architecture of the FTC block in the F2LM generator. Abbreviations: *k*: kernel, *p*: padding, *s*: stride, *H*: height, *W*: width, *C*: channel.

Layer name	Layer	Input size $(H \times W \times C)$	Output size $(H \times W \times C)$
doubleconv	[Conv2D (*k*=3, *p*=1,*s*=1) BatchNorm2d ReLU] × 2	$32 \times 32 \times 512$	$32 \times 32 \times 512$
LN	L_2 normalization	$32 \times 32 \times 512$	$32 \times 32 \times 512$

TABLE 4. AUC comparison based on the feature fusion method. Best results are bolded.

Fusion method	UCSD Ped2	CUHK Avenue	Shanghai Tech.
Add	97.3%	87.5%	72.6%
Concat & SE block	**97.5%**	**88.2%**	**74.3%**

TABLE 5. AUC comparison based on the hyperparameter α of the triplet loss. Best results are bolded.

Triplet α	UCSD Ped2	CUHK Avenue	Shanghai Tech.
0.2	**97.5%**	**88.2%**	**74.3%**
0.5	97.0%	87.5%	72.4%
0.8	95.9%	87.0%	72.4%
1.0	95.6%	86.7%	72.3%

TABLE 6. AUC comparison based on the dropout noise method. Abbreviations: CD: channel dependent, CI: channel independent. Best results are bolded.

Noise method	UCSD Ped2	CUHK Avenue	Shanghai Tech.
None	97.6%	90.0%	75.1%
Dropout(CD)	97.8%	90.6%	75.7%
Dropout(CI)	**98.2%**	**91.2%**	**76.5%**

structures of the F2LM generator are presented in Table 1 and Table 2, respectively. The input video sequence consists of four frames, DeepLabv3 generates information for 21 classes for each frame, whereas FlowNet2 generates flow maps for x and y coordinates between adjacent frames.

TABLE 7. AUC comparison according to patch size. Best results are bolded.

Patch size	UCSD Ped2	CUHK Avenue	Shanghai Tech.
16×16	97.8%	90.4%	75.8%
32×32	**98.2%**	**91.2%**	**76.5%**
64×64	97.7%	89.8%	76.1%
128×128	97.6%	89.7%	75.9%

TABLE 8. AUC comparison based on the hyperparameter λ of the Destroyer. Best results are bolded.

λ	UCSD Ped2	CUHK Avenue	Shanghai Tech.
1	97.5%	90.2%	75.6%
2	97.6%	90.4%	76.1%
3	97.8%	91.0%	76.3%
4	**98.2%**	**91.2%**	**76.5%**
5	98.0%	90.7%	76.2%
6	98.0%	90.6%	76.3%

TABLE 9. AUC comparison based on the selection of Z_p. Best results are bolded.

Z_p	UCSD Ped2	CUHK Avenue	Shanghai Tech.
None	97.5%	88.2%	74.3%
Background vector	97.6%	91.0%	-
Zero vector	**98.2%**	**91.2%**	**76.5%**

TABLE 10. Hyperparameters for the entire network.

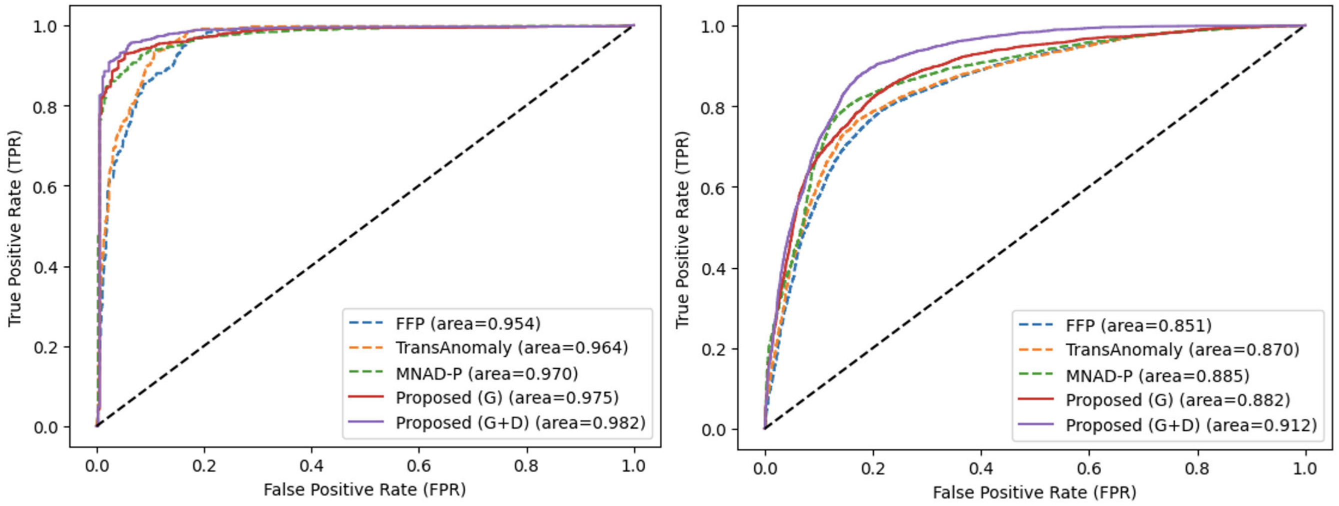
Hyper-P	Value	Description
α	0.2	Margin of triplet loss
noise	CI	Noising method for the Destroyer training
patch size	32	Size of patch for patching
λ	4	Number for adjusting $diff_p$
Z_p	zero vector	Destroying method
δ_{int}	1	Weight of intensity loss
δ_{gd}	1	Weight of gradient loss
δ_{adv}	0.05	Weight of adversarial loss
δ_{tri}	1	Weight of triplet loss
γ_1	(0.02, 0.94, 0.68)	Weight of label triplet loss for testing
γ_2	(0.50, 0.02, 0.06)	Weight of motion triplet loss for testing
γ_3	(0.48, 0.04, 0.26)	Weight of F2LM generator MSE loss for testing
γ_4	(1.00, 1.00, 0.25)	Weight of Destroyer MSE loss for testing

Therefore, the values of channel (C) for each encoder E_f, E_l, and E_m are 12, 84, and 6, respectively. In the final layer of the encoder, L_2 normalization is applied to the output of the encoder to calculate the triplet loss. The structure of the FTC block is presented in Table 3. For the same reason as for the encoder of the F2LM generator, L_2 normalization is applied to the last layer.

We investigate the performance variation of the F2LM generator based on different feature fusion methods; the results are presented in Table 4. Although Z_f, Z_l', Z_m' can be simply added and used as input to the decoder, we opt for channel-wise concatenation and the SE block for channel attention. We observe that utilizing the SE block for feature fusion, which can capture important information from the three features, contributed to performance improvement. Therefore, we employ the SE block instead of simply adding the three features.

TABLE 11. AUC and EER comparison with state-of-the-art methods using the UCSD Ped2, CUHK Avenue, and Shanghai Tech. datasets. Best results are bolded. Second-best results are underlined. w/o Destroyer: without Destroyer.

Year	Method	UCSD Ped2		CUHK Avenue		Shanghai Tech.		Publisher
		AUC	EER	AUC	EER	AUC	EER	
2018	FFP [10]	95.4%	11.7%	85.1%	21.4%	72.8%	33.1%	CVPR
	Wang et al. [38]	96.4%	8.9%	85.3%	23.9%	-	-	MM
2019	MemAE [6]	94.1%	-	83.3%	-	71.2%	-	ICCV
	AMC [8]	96.2%	-	86.9%	-	-	-	ICCV
	AnoPCN [11]	96.8%	-	86.2%	-	73.6%	-	MM
	AnomalyNet [39]	94.9%	10.3%	86.1%	22.0%	-	-	IEEE Transactions
2020	DSTN [40]	95.5%	9.4%	87.9%	20.2%	-	-	IEEE Access
	Siamese [41]	94.0%	14,1%	-	-	-	-	WACV
	GMM-FCN [42]	92.2%	12.6%	83.4%	22.7%	-	-	CVIU
	Tang et al. [43]	96.3%	10.0%	85.1%	-	73.0%	-	Pattern Recognition Letters
	FFP+MS_SSIM+FCN [44]	95.9%	11.1%	85.9%	20.4%	73.5%	32.5%	ICCC
	Dual D-b GAN [45]	95.6%	-	84.9%	-	73.7%	32.2%	IEEE Access
	MNAD-P [5]	97.0%	-	88.5%	-	70.5	-	CVPR
2021	TransAnomaly [12]	96.4%	-	87.0%	-	-	-	IEEE Access
	BR-GAN [21]	97.6%	7.6%	88.6%	19.0%	74.5%	31.6%	IEEE Access
	Multi-scale U-Net [46]	95.7%	12.0%	86.9%	20.2%	73.0%	32.3%	IEEE Access
	HMCF [47]	93.7%	18.8%	83.2%	20.0%	-	-	MobileHCI
	HF2-VAD [17]	**99.3%**	-	91.1%	-	76.2%	-	ICCV
2022	Zhong et al. [9]	97.7%	-	88.9%	-	70.7%	-	Pattern Recognition
	DLAN-AC [16]	97.6%	-	89.9%	-	74.7%	-	ECCV
	DR-STN [48]	97.6%	6.9%	90.8%	**11.0%**	-	-	Pattern Recognition Letters
2023	Scene-Aware [49]	-	-	89.6%	21.1%	74.7%	**28.6%**	MM
	MsMp-net [50]	97.6%	6.6%	89.0%	18.1%	-	-	IEEE Access
	Bi-READ [51]	97.7%	7.9%	86.7%	19.5%	-	-	VCIR
	USTN-DSC [52]	98.1%	-	89.9%	-	73.8%	-	CVPR
	SwinAnomaly [53]	98.2%	-	84.8%	-	76.3%	-	IEEE Access
Proposed	**Ours (w/o Destroyer)**	97.5%	7.0%	88.2%	19.1%	74.3%	31.4%	
	Ours	98.2%	**5.9%**	**91.2%**	15.5%	**76.5%**	30.0%	



(a) UCSD Ped2

(b) CUHK Avenue

FIGURE 7. ROC comparison at the frame level for the UCSD Ped2 and CUHK Avenue datasets.

b: DESTROYER

The U-Net encoder used in the Destroyer has a structure similar to that of the encoder of the F2LM generator in Table 1. However, in the *inconv* layer, *C* is three, and L_2 normalization is not applied after the *downconv*$_3$ layer. The U-Net decoder used in the Destroyer is the same as the decoder structure of the F2LM generator in Table 2.

TABLE 12. Performance results at the frame and pixel levels for the UCSD Ped2 dataset.

Method	Frame level		Pixel level	
	AUC	EER	AUC	EER
Wang et al. [38]	96.4%	8.9%	85.9%	19.4%
Siamese [41]	94.0%	14.1%	**93.0%**	-
DSTN [40]	95.5%	9.4%	83.1%	21.8%
GMM-FCN [42]	92.2%	12.6%	78.2%	19.2%
DR-STN [48]	97.6%	6.9%	86.4%	16.3%
Ours (w/o Destroyer)	97.5%	7.0%	90.3%	16.4%
Ours	**98.2%**	**5.9%**	92.3%	**16.2%**

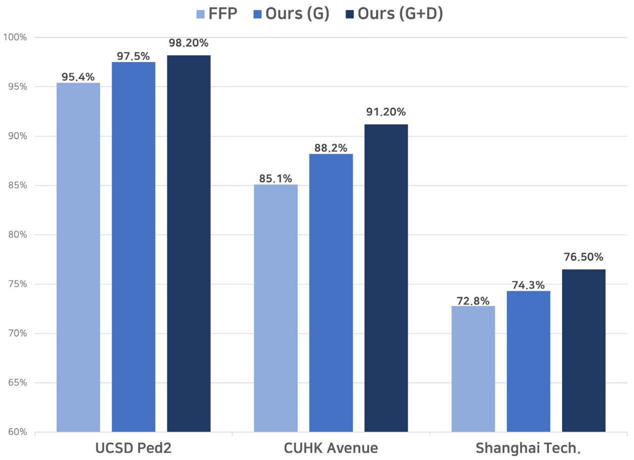


FIGURE 8. AUC comparison with baseline [10]. Abbreviations: FFP: future frame prediction, G: F2LM generator, D: Destroyer.

2) HYPERPARAMETER SETTINGS

We conduct five experiments for hyperparameters.

First, we examine the performance of the F2LM generator with respect to the hyperparameter α in the triplet loss. The results of this experiment are shown in Table 5. The parameter α is used to control the distance between the anchor and the negative feature; a larger α encourages learning a larger distance between them. We vary α from 0.2 to 1 in increments of 0.2. Through the experiment, we confirm that the highest area under the curve (AUC) is recorded for all three datasets when α is set to 0.2.

Second, we examine the performance changes of the Destroyer based on the method of adding noise; the results are presented in Table 6. Based on the experimental results, we select the dropout method used in the denoising autoencoder [54] as the noise adding method. We experiment with two noise adding methods: channel dependent (CD), which applies noise to all R, G, and B channels in the same ratio, and channel independent (CI), which applies noise with different ratios to each channel. The experimental results confirm the CI method is more effective for video anomaly detection. Furthermore, by comparing the performance when applying CD or CI to the case without noise, we observe that performance is enhanced in both cases. This verifies the effectiveness of self-supervised learning in the Destroyer.

Third, we examine the model's performance variation based on patch size; the results are presented in Table 7. We use patch sizes ranging from 16 to 128, doubling in each step, and find that the model achieves the best performance for all datasets with a patch size of 32.

Fourth, we investigate performance variation with respect to the hyperparameter λ in the Destroyer loss; the results are presented in Table 8. λ is used to control for the $diff_p$ value in (5). Even when the quality difference between N_p and R_p is small, indicating a high SSIM value, if λ is large, learning proceeds toward destruction. We explore the values of λ from 1 to 6 and find that the highest AUC for all three datasets is achieved when λ is set to 4.

Fifth, we conduct a comparative experiment with the zero and background vectors as shown in Table 9 to set Z_p for the Destroyer learning. The experiments confirm that the highest AUC is obtained for the three benchmark datasets when the zero vector is used. The background vector is calculated by averaging the pixels of the entire frame.

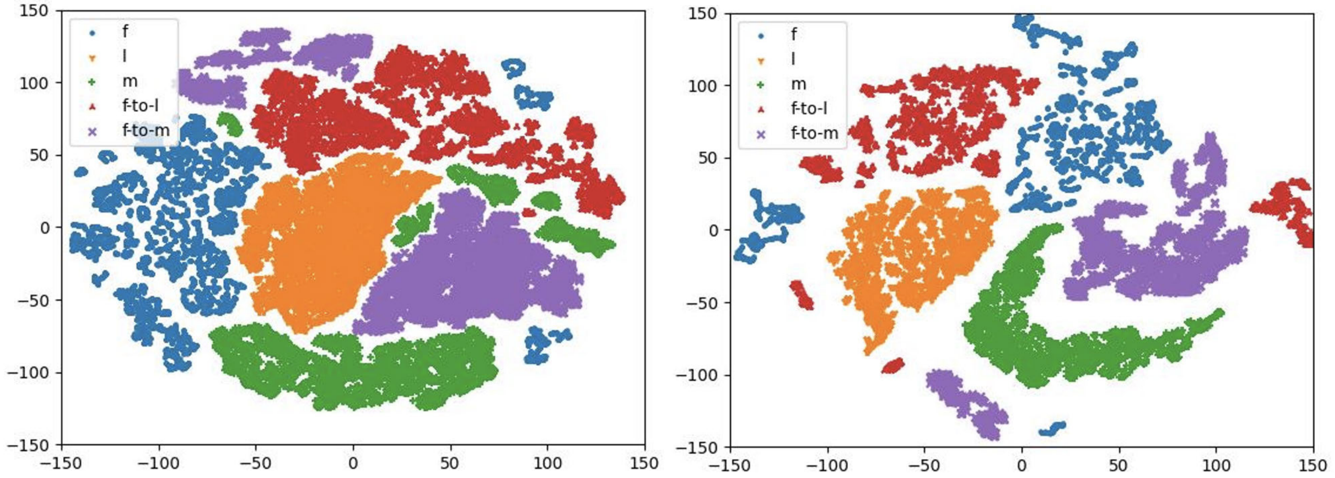
Finally, the hyperparameters of the entire network are summarized in Table 10. δ_{int}, δ_{gd}, δ_{adv}, δ_{tri} are hyperparameters that adjust the influence of each term in the F2LM generator loss. $\gamma_1, \gamma_2, \gamma_3$, and γ_4 are hyperparameters that control for the influence of each term in the scaled score. These hyperparameters differ for each dataset, and the values presented in Table 10 are in the order of UCSD Ped2, CUHK Avenue, and Shanghai Tech. datasets.

C. EVALUATION CRITERIA

We evaluate the quantitative performance of the proposed model based on two criteria: frame and pixel levels. At the frame level, a test frame is considered anomalous if it contains one or more abnormal events. By contrast, the pixel level specifies the locations of abnormal events. Pixel level evaluation is more challenging than frame level evaluation owing to the complexity of anomaly localization. The frame level and pixel level AUC performance comparisons for the UCSD Ped2 dataset are presented in Table 12. In the case of the CUHK Avenue dataset, because the ground truth of the abnormal area of each frame is set as a rectangular bounding box, there is a problem in that the abnormal area of the ground truth includes not only foreground but also background pixels. Therefore, we ignore pixel level measurements for the CUHK Avenue dataset and used only frame level measurements for testing [42].

D. COMPARISON WITH STATE-OF-THE-ART MODELS

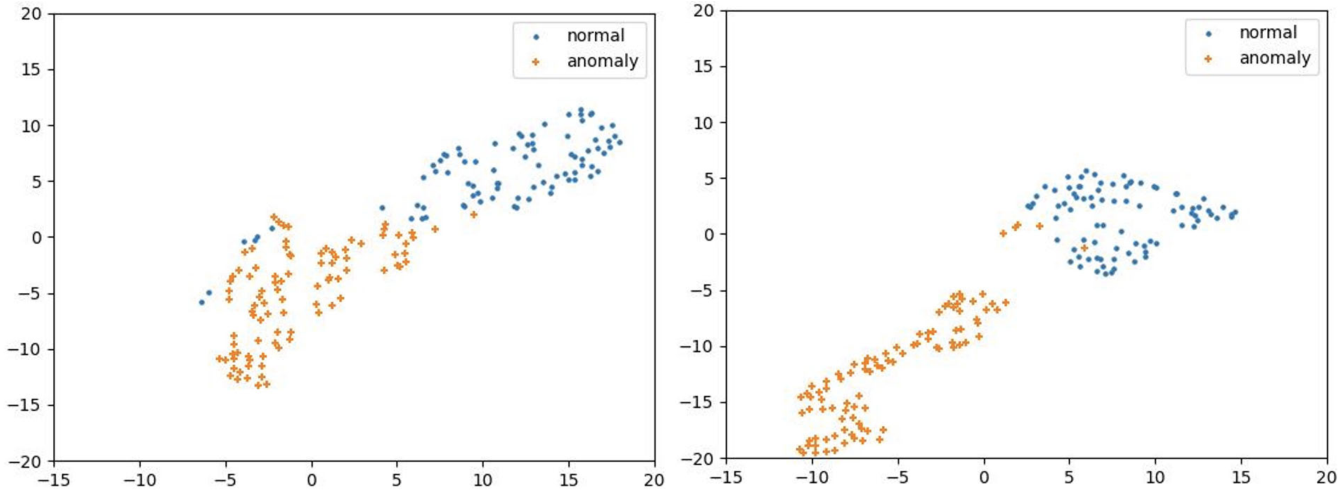
Table 11 presents the comparison of our model to other deep learning-based models. We gradually adjust the threshold value of the anomaly score for existing methods that utilize frame level prediction methods and for our proposed model to generate a receiver operating characteristic (ROC) curve, as shown in Fig. 7. We also calculat the AUC and equal error rate (EER) for performance evaluation.



(a) Normal features

(b) Anomaly features

FIGURE 9. t-SNE results of normal and anomaly features generated by the F2LM generator on the CUHK Avenue dataset. Abbreviations: f: features Z_f extracted from encoder E_f, l: features Z_l extracted from encoder E_l, m: features Z_m extracted from encoder E_m, f-to-l: features Z'_l transformed from FTC block, f-to-m: features Z'_m transformed from FTC block.



(a) The F2LM generator

(b) The Destroyer

FIGURE 10. t-SNE results for output features of the F2LM generator and Destroyer for the UCSD Ped2 dataset. Abbreviations: normal: features of normal frames, anomaly: features of anomaly frames.

TABLE 13. AUC comparison based on combinations of frame, label, and motion encoders (E_f, E_l, E_m) and FTC block (*FTC*). Best results are **bolded**.

E_f	E_l	E_m	*FTC*	UCSD Ped2	CUHK Avenue	Shanghai Tech.
√				95.4%	85.1%	72.8%
√	√		√	95.8%	85.7%	73.0%
√		√	√	96.8%	86.2%	73.4%
√	√	√	√	**97.5%**	**88.2%**	**74.3%**
√	√	√		95.2%	83.2%	71.0%

In terms of AUC performance, our proposed method has a better AUC than existing state-of-the-art methods except for the HF2-VAD [17] model using an object detection method; particularly, it achieves state-of-the-art AUC for the CUHK Avenue and Shanghai Tech. datasets. Additionally, as can be seen from the pixel level comparison in Table 12, the AUC for the UCSD Ped2 dataset is superior to that of the other methods, except for the Siamese [41] model. The Siamese's approach appears to achieve a high AUC at the pixel level when using supervised learning based on labeled data [48]. However, our experimental results show competitive performance in anomaly detection and localization tasks despite using unsupervised learning methods and not applying memory modules or object detection methods.

Fig. 8 illustrates the performance comparison between our model and the baseline, future frame prediction [10]. Compared with the baseline, the F2LM generator showed improvements of 2.1%, 3.1%, and 1.5% for the UCSD Ped2, CUHK Avenue, and Shanghai Tech. datasets. This indicates the effectiveness of utilizing the label and motion information. When using the Destroyer in conjunction, performance improvements of 2.8%, 6.1%, and 3.7% were achieved for the UCSD Ped2, CUHK Avenue, and Shanghai Tech. datasets.

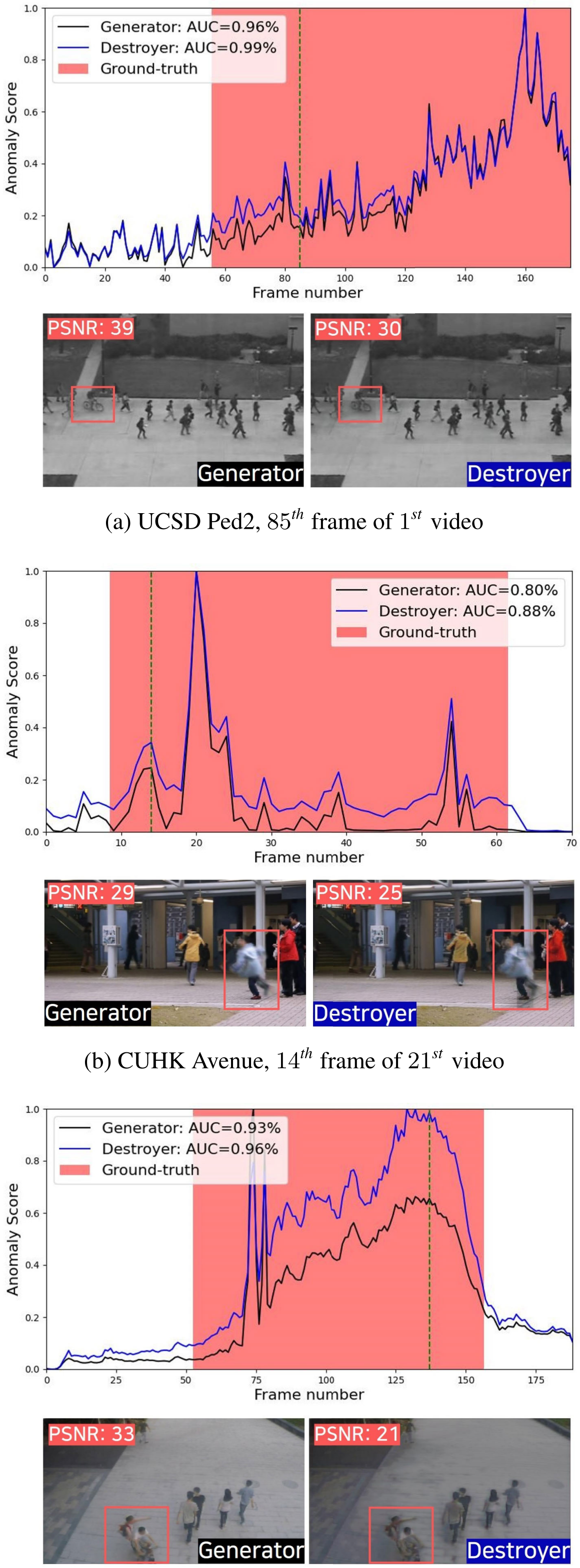
This demonstrates that the Destroyer effectively enhances performance by increasing the anomaly score difference between normal and abnormal data through the destruction of abnormal regions. In particular, we confirm that the Destroyer can better distinguish abnormal areas in datasets such as the CUHK Avenue, where occlusions are less frequent, and successfully destroy abnormal areas.

E. QUALITATIVE EVALUATION

1) NORMAL AND ABNORMAL FEATURE VISUALIZATION

We visualize how the F2LM generator transforms features for label and motion, representing normal and abnormal features, using t-SNE in Fig. 9. In the visualization of normal data, as shown in Fig. 9s, we observe that the frame-to-motion(f-to-m) feature (depicted as purple ‘‘x’’) is closer to the positive motion feature (m) (depicted as green ‘‘+’’), than to the negative frame feature (f) (depicted as blue ‘‘·’’). However, in the visualization of anomalous data, we observe that the distance between f-to-m and m is similar to that between f-to-m and f, as shown in Fig. 9b. Similarly, a pattern is observed in which the distance between f-to-l and l is similar to that between f-to-l and f. This indicates that the F2LM generator performs well for feature transformation for normal data but struggles for abnormal data. Consequently, features converted from frames to labels and motions act as noise for anomalous inputs, resulting in low-quality future frames.

Furthermore, we visualize the output features of the F2LM generator and Destroyer to confirm whether the Destroyer effectively destroys abnormal regions within the future frame generated by the F2LM generator, making it easier to distinguish between normal and abnormal frames. This process is illustrated in Fig. 10. The F2LM generator shown in Fig. 10a distinguishes between normal and abnormal clusters well, but the distance between the two clusters is relatively close compared to that from the Destroyer. For the Destroyer, as shown in Fig. 10b, it is evident that the two clusters are better separated, resulting in a greater distance between them. This suggests that the Destroyer enhances the discrimination between normal and abnormal regions by effectively destroying abnormal regions, leading to a clearer separation between the clusters.



(a) UCSD Ped2, 85th frame of 1st video

(b) CUHK Avenue, 14th frame of 21st video

(c) Shanghai Tech., 137th frame 32nd video

FIGURE 11. Visualization results of the F2LM generator and Destroyer for each dataset. The red regions represent ground truth abnormal frames. This indicates that the Destroyer lowers the PSNR of abnormal frames, resulting in a higher anomaly score. Best viewed in color.

TABLE 14. AUC results when applying our Destroyer to generators in existing methods. To ensure consistent model comparisons, we did not employ the sliding windows strategy for calculating AUC. w/ Destroyer: with Destroyer, w/o Destroyer: without Destroyer.

Method	Generator	Destroyer	UCSD Ped2	CUHK Avenue	Shanghai Tech.
Frame-Prediction [10]	√		95.4%	84.9%	72.5%
Frame-Prediction (w/ Destroyer)	√	√	96.3%(+0.9%)	87.7%(+2.4%)	72.7%(+0.2%)
TransAnomaly [12]	√		96.2%	85.6%	N/A
TransAnomaly (w/ Destroyer)	√	√	96.8%(+0.6%)	88.2%(+2.6%)	N/A
Ours (w/o Destroyer)	√		97.5%	88.2%	74.3%
Ours	√	√	98.2%(+0.7%)	91.2%(+3.0%)	76.5%(+2.2%)

TABLE 15. Comparison of computational time during testing (seconds per frame). w/o Destroyer: without Destroyer.

Method	CPU	GPU	Memory	Average Running Time
FFP [10]	3.4GHz	Nvidia GeForce TITAN	-	0.040
DSTN [40]	2.8GHz	Nvidia GeForce GTX 1080 Ti	24GB	0.321
BR-GAN [21]	3.8GHz	Nvidia GeForce RTX 2070 super	24GB	0.022
Multi-scale U-Net [46]	-	Nvidia GeForce RTX 2080 Ti	24GB	0.041
TransAnomaly [12]	-	Nvidia RTX 3070	-	0.056
MsMp-net [50]	2.3GHz	Nvidia GeForce GTX 1080 Ti	-	0.089
Ours(w/o Destroyer)	3.7GHz	Nvidia GeForce RTX 3090	64GB	0.047
Ours	3.7GHz	Nvidia GeForce RTX 3090	64GB	0.052

2) VISUALIZATION ANOMALY SCENE

In Fig. 11, the AUCs of the F2LM generator and Destroyer are compared, and visualizations are presented for specific frames. When comparing anomaly scores, those from the Destroyer are higher than those from the F2LM generator. In particular, there is a significant difference in the anomaly scores between normal and abnormal frames, making the discrimination between normal and abnormal frames easier and resulting in improved AUC performance. Furthermore, the Destroyer is observed to destruct areas around abnormal objects. In Fig. 11a, the area where a person is riding a bicycle is identified as destructed compared to the frame generated by the F2LM generator, and the PSNR also decreases from 39 to 30. In Fig. 11b, the abnormal behavior area where a child is jumping is destructed, and the PSNR decreases from 29 to 25. In Fig. 11c, the area where two people are riding bicycles is destructed, and the PSNR decreases from 33 to 21. This indicates that the Destroyer enhances video anomaly detection performance by destroying abnormal regions.

F. ABLATION STUDY

1) F2LM GENERATOR

Traditional anomaly detection methods have primarily focused on predicting future frames by utilizing information from preceding frames. However, we assume that when utilizing the encoder to extract additional label and motion information of objects within video frames and employing the FTC block, it would be challenging to generate future frames when abnormal frames are input. To validate this assumption, we conduct a comparative evaluation using various architectures, as shown in Table 13. Consequently, we confirm that leveraging all three pieces of information while utilizing the FTC block is effective in enhancing the performance of the anomaly detection model. Furthermore, we observe that when all three pieces of information are utilized without the FTC block, the additional information regarding labels and motion results in a superior generation of future frames for both normal and abnormal scenarios, leading to diminished anomaly detection performance. This finding substantiates the validity of our assumption.

2) DESTROYER

We examine the generalization performance of the Destroyer by assessing its performance using various generators, as illustrated in Table 14. We utilize future frame prediction [10], which is a representative prediction-based approach, and TransAnomaly [12], which employs ViViT, as generators. We train the future frame prediction model using the official code, whereas for TransAnomaly, we use our own code because no official code is provided. The results show the proposed Destroyer improves performance across all datasets. In particular, the CUHK Avenue dataset, which has fewer occlusions, shows significant overall performance improvement. However, for the Shanghai Tech. dataset, the performance improvement is not significant compared to that of our proposed F2LM generator. This suggests that the F2LM generator may not accurately predict abnormal regions, increasing the effectiveness of the Destroyer. These results imply that the proposed F2LM generator is better suited for use with the Destroyer.

G. RUNNING TIME

Table 15 presents the analysis of the computational time of our proposed model on a graphics processing unit and compares it with that of state-of-the-art methods. For the UCSD Ped2 and CUHK Avenue datasets, the average computational times of the proposed model without and with the Destroyer is 0.047 seconds (21 FPS), and 0.052 seconds (19 FPS), respectively. Our proposed model is equivalent to or slightly faster than DSTN [40], TransAnomaly [12], and MsMp-net [50] and exceeds the inference time of FFP [10], BR-GAN [21], and Multi-scale U-Net [46]. Our proposed model requires a longer average computational time because it is a two-stage method that sequentially executes the F2LM generator, which comprises three encoders, and the Destroyer. However, as shown in Table 11, better results are obtained compared to the other models from an AUC perspective.

V. CONCLUSION

We proposed a novel video anomaly detection method that utilizes the F2LM generator to predict low-quality future frames for abnormal video sequences and the Destroyer to destroy low-quality areas. In this approach, the FTC block in the F2LM generator is trained for feature transformation using only normal video sequences, resulting in challenges for feature transformation for abnormal video sequences, which decreases the quality of predicted future frames. Subsequently, the Destroyer identifies low-quality areas in the future frame predicted by the F2LM generator and destroys them into zero vectors, making them more anomalous. This approach addresses the limitations of previous prediction-based and reconstruction-based methods and shows superior performance across all benchmark datasets. These results demonstrate the effectiveness of the proposed video anomaly detection method. We anticipate the development of various techniques to destroy abnormal regions.

ACKNOWLEDGMENT

(Seungkyun Hong and Sunghyun Ahn are co-first authors.)

REFERENCES

[1] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 733–742.

[2] W. Luo, W. Liu, and S. Gao, "Remembering history with convolutional LSTM for anomaly detection," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2017, pp. 439–444.

[3] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Proc. Int. Symp. Neural Netw.*, May 2017, pp. 189–196.

[4] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 341–349.

[5] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14360–14369.

[6] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. Van Den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1705–1714.

[7] Y. Chang, Z. Tu, W. Xie, and J. Yuan, "Clustering driven deep autoencoder for video anomaly detection," in *Computer Vision—ECCV*. Cham, Switzerland: Springer, 2020, pp. 329–345.

[8] T. N. Nguyen and J. Meunier, "Anomaly detection in video sequence with appearance-motion correspondence," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1273–1283.

[9] Y. Zhong, X. Chen, J. Jiang, and F. Ren, "A cascade reconstruction model with generalization ability evaluation for anomaly detection in videos," *Pattern Recognit.*, vol. 122, Feb. 2022, Art. no. 108336.

[10] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[11] M. Ye, X. Peng, W. Gan, W. Wu, and Y. Qiao, "AnoPCN: Video anomaly detection via deep predictive coding network," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1805–1813.

[12] H. Yuan, Z. Cai, H. Zhou, Y. Wang, and X. Chen, "TransAnomaly: Video anomaly detection using video vision transformer," *IEEE Access*, vol. 9, pp. 123977–123986, 2021.

[13] R. Cai, H. Zhang, W. Liu, S. Gao, and Z. Hao, "Appearance-motion memory consistency network for video anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2021, pp. 938–946.

[14] H. Wang, X. Zhang, S. Yang, and W. Zhang, "Video anomaly detection by the duality of normality-granted optical flow," 2021, *arXiv:2105.04302*.

[15] H. Lv, C. Chen, Z. Cui, C. Xu, Y. Li, and J. Yang, "Learning normal dynamics in videos with meta prototype network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 15420–15429.

[16] Z. Yang, P. Wu, J. Liu, and X. Liu, "Dynamic local aggregation network with adaptive clusterer for anomaly detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 404–421.

[17] Z. Liu, Y. Nie, C. Long, Q. Zhang, and G. Li, "A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13568–13577.

[18] M. Zaigham Zaheer, J.-H. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14171–14181.

[19] C. Huang, J. Wen, Y. Xu, Q. Jiang, J. Yang, Y. Wang, and D. Zhang, "Self-supervised attentive generative adversarial networks for video anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, pp. 9389–9403, 2022.

[20] C. Huang, Z. Wu, J. Wen, Y. Xu, Q. Jiang, and Y. Wang, "Abnormal event detection using deep contrastive learning for intelligent video surveillance system," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5171–5179, Aug. 2022.

[21] Z. Yang, J. Liu, and P. Wu, "Bidirectional retrospective generation adversarial network for anomaly detection in videos," *IEEE Access*, vol. 9, pp. 107842–107857, 2021.

[22] M. Astrid, M. Zaigham Zaheer, J.-Y. Lee, and S.-I. Lee, "Learning not to reconstruct anomalies," 2021, *arXiv:2110.09742*.

[23] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.

[24] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[25] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1655.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351, 2015, pp. 234–241.

[28] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[29] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.

[30] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[31] N.-C. Ristea, N. Madan, R. T. Ionescu, K. Nasrollahi, F. S. Khan, T. B. Moeslund, and M. Shah, "Self-supervised predictive convolutional attentive block for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13566–13576.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821.

[34] A. Chan and N. Vasconcelos, "UCSD pedestrian dataset," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, pp. 909–926, 2008.

[35] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2720–2727.

[36] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[37] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[38] S. Wang, Y. Zeng, Q. Liu, C. Zhu, E. Zhu, and J. Yin, "Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 636–644.

[39] J. T. Zhou, J. Du, H. Zhu, X. Peng, Y. Liu, and R. S. M. Goh, "AnomalyNet: An anomaly detection network for video surveillance," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 10, pp. 2537–2550, Oct. 2019.

[40] T. Ganokratanaa, S. Aramvith, and N. Sebe, "Unsupervised anomaly detection and localization based on deep spatiotemporal translation network," *IEEE Access*, vol. 8, pp. 50312–50329, 2020.

[41] B. Ramachandra, M. J. Jones, and R. Raju Vatsavai, "Learning a distance function with a Siamese network to localize anomalies in videos," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 2587–2596.

[42] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, "Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder," *Comput. Vis. Image Understand.*, vol. 195, Jun. 2020, Art. no. 102920.

[43] Y. Tang, L. Zhao, S. Zhang, C. Gong, G. Li, and J. Yang, "Integrating prediction and reconstruction for anomaly detection," *Pattern Recognit. Lett.*, vol. 129, pp. 123–130, Jan. 2020.

[44] Y. Yang, D. Zhan, F. Yang, X.-D. Zhou, Y. Yan, and Y. Wang, "Improving video anomaly detection performance with patch-level loss and segmentation map," in *Proc. IEEE 6th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2020, pp. 1832–1839.

[45] F. Dong, Y. Zhang, and X. Nie, "Dual discriminator generative adversarial network for video anomaly detection," *IEEE Access*, vol. 8, pp. 88170–88176, 2020.

[46] S. Saypadith and T. Onoye, "An approach to detect anomaly in video using deep generative network," *IEEE Access*, vol. 9, pp. 150903–150910, 2021.

[47] F. Yang, Z. Yu, L. Chen, J. Gu, Q. Li, and B. Guo, "Human-machine cooperative video anomaly detection," *Proc. ACM Hum.-Comput. Interact.*, vol. 4, no. CSCW3, pp. 1–18, Jan. 2021.

[48] T. Ganokratanaa, S. Aramvith, and N. Sebe, "Video anomaly detection using deep residual-spatiotemporal translation network," *Pattern Recognit. Lett.*, vol. 155, pp. 143–150, Mar. 2022.

[49] C. Sun, Y. Jia, Y. Hu, and Y. Wu, "Scene-aware context reasoning for unsupervised abnormal event detection in videos," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 184–192.

[50] N. Taghinezhad and M. Yazdi, "A new unsupervised video anomaly detection using multi-scale feature memorization and multipath temporal information prediction," *IEEE Access*, vol. 11, pp. 9295–9310, 2023.

[51] R. Kommanduri and M. Ghorai, "Bi-READ: Bi-residual AutoEncoder based feature enhancement for video anomaly detection," *J. Vis. Commun. Image Represent.*, vol. 95, Sep. 2023, Art. no. 103860.

[52] Z. Yang, J. Liu, Z. Wu, P. Wu, and X. Liu, "Video event restoration based on keyframes for video anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14592–14601.

[53] A. Bajgoti, R. Gupta, P. Balaji, R. Dwivedi, M. Siwach, and D. Gupta, "SwinAnomaly: Real-time video anomaly detection using video Swin transformer and SORT," *IEEE Access*, vol. 11, pp. 111093–111105, 2023.

[54] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

SEUNGKYUN HONG received the B.S. degree in physics and the M.S. and Ph.D. degrees in computer science from Yonsei University, Seoul, South Korea, in 1993 and 2018, respectively. He was an Information Technology Expert with SK Telecom and POSCO, and worked in the industry for 30 years. His research interests include machine learning, anomaly detection, image segmentation, and neural networks.

SUNGHYUN AHN received the B.S. degree in computer science and information engineering from Catholic University, South Korea, in 2023. He is currently pursuing the M.S. degree in computer science with Yonsei University, Seoul, South Korea. His current research interests include deep learning, computer vision, and image understanding.

YOUNGWAN JO received the B.S. degree in computer science from Kookmin University, Seoul, South Korea, in 2022 and the M.S. degree in computer science with Yonsei University, Seoul. His current research interests include deep learning, computer vision, anomaly detection, and medical image segmentation.

SANGHYUN PARK (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Seoul National University, in 1989 and 1991, respectively, and the Ph.D. degree from the Department of Computer Science, University of California at Los Angeles (UCLA), in 2001. He is currently a Professor with the Department of Computer Science, Yonsei University, Seoul, South Korea. His current research interests include databases, data mining, bioinformatics, and flash memory.