IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 11, 2020, accepted February 29, 2020, date of publication March 10, 2020, date of current version March 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2979869

Unsupervised Anomaly Detection and Localization Based on Deep Spatiotemporal Translation Network

THITTAPORN GANOKRATANAA[1], (Graduate Student Member, IEEE), SUPAVADEE ARAMVITH[1], (Senior Member, IEEE), AND NICU SEBE[2], (Senior Member, IEEE)

[1]Department of Electrical Engineering, Chulalongkorn University, Bangkok 10330, Thailand
[2]Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy

Corresponding author: Supavadee Aramvith (supavadee.a@chula.ac.th)

This work was supported by the Chulalongkorn University Dutsadi Phiphat Scholarship.

ABSTRACT Anomaly detection is of great significance for intelligent surveillance videos. Current works typically struggle with object detection and localization problems due to crowded and complex scenes. Hence, we propose a Deep Spatiotemporal Translation Network (DSTN), novel unsupervised anomaly detection and localization method based on Generative Adversarial Network (GAN) and Edge Wrapping (*EW*). In training, we use only the frames of normal events in order to generate their corresponding dense optical flow as temporal features. During testing, since all the video sequences are input into the system, unknown events are considered as anomalous events due to the fact that the model knows only the normal patterns. To benefit from the information provided by both appearance and motion features, we introduce (i) a novel fusion of background removal and real optical flow frames with (ii) a concatenation of the original and background removal frames. We improve the performance of anomaly localization in the pixel-level evaluation by proposing (iii) the Edge Wrapping to reduce the noise and suppress non-related edges of abnormal objects. Our DSTN has been tested on publicly available anomaly datasets, including UCSD pedestrian, UMN, and CUHK Avenue. The results show that it outperforms other state-of-the-art algorithms with respect to the frame-level evaluation, the pixel-level evaluation, and the time complexity for abnormal object detection and localization tasks.

INDEX TERMS Anomaly detection, anomaly localization, spatiotemporal, unsupervised learning, video surveillance.

I. INTRODUCTION

Surveillance has rapidly gained increasing popularity as a modern technology, which can be used to ensure life safety and break the wall of security mistrust. Closed-Circuit Television (CCTV) cameras have been widely used for monitoring and recording situations, providing evidence to the surveillance system. According to [1], the growth of surveillance videos has increased by 9.3 percent in 2019. However, the CCTV cameras are mostly used for the post-video forensic process by allowing the investigation of previous events [2]. This means that the CCTV camera feed still needs to be manually monitored by a human operator for any abnormal events which can unpredictably occur in the scene. An abnormal or anomalous event refers to an activity that raises suspicions by differing from the majority of the activities. It can possibly occur in any realistic scenario (e.g. indoor, outdoor, crowded, and uncrowded scenes) and may lead to major problems, such as an area invasion, a robbery, and a terrorist attack, causing a lot of damage, injury, or death [3]. According to the performance of CCTV cameras [2], there is a need to build intelligent systems to analyze abnormal events in realistic scenes for surveillance videos. The main challenge of building an intelligent CCTV system is how to precisely detect and locate all possible abnormal events in crowded and complex scenes.

To design an effective anomaly detection and localization system [7], [10], [13], [44], there are four main issues

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

to be considered: the complex scene, time-consumption, dataset, and object localization. The complex or crowded scene may contain multiple objects with clutter and occlusions which are difficult to deal with. Besides, it is more challenging than the uncrowded scene as it has higher complexity. This scene complexity challenge has drawn interest from researchers in computer vision research area [4]–[14], [19], [20], [45]–[48]. To handle this significant issue, two main approaches have been implemented for anomaly detection in crowds: (i) a traditional-based approach and (ii) a deep learning-based approach. With (i) the traditional-based approaches [20]–[28], [35], appearance and motion (e.g. trajectories) are employed to detect the anomaly events based on hand-crafted features. Their accuracy depends on object appearances and motion cues which can be found by extracting features and tracking the objects [20]. Even though the traditional-based approaches are able to detect multiple objects in crowded scenes, they are more difficult to generalize to complex scenarios than deep learning-based approaches. Hence, deep learning-based approaches [4], [7], [10]–[17], [29], have been considered as being more appropriate for handling complex scenes as they are able to improve the performance of anomaly detection and localization with the use of a learnable model of nonlinear transformation [7], [8], [13], [15]. Following the complex scene issue, time-consumption is one of the challenging issues for the use of an anomaly detection system in real-world applications. If high accuracy is required, the detection of multiple objects in crowded scenes is very time-consuming, asking for an inherent speed-accuracy tradeoff [5], [18], [20], [38]–[41]. Recently, the deep learning-based approaches were considered for reducing the time complexity while retaining good detection performance due to the importance of low computational complexity and high detection accuracy for the surveillance videos [44], [46], [52]–[57], [59]. The recent advanced techniques for speeding up CNNs are parameter pruning and sharing and transferred convolutional filters [30]. Many works [52]–[57] try to optimize the computational time of CNN-based algorithms, focusing on convolutional architectures by reducing convolutional layers and redundant parameters that are not drastically impacting the model performance, resulting in a smaller and faster network compared to the traditional CNN [58]. Several works [46], [59] use pre-trained fully convolutional networks (FCNs) as a regional feature extractor for semantic segmentation to help to reduce the computational complexity of the traditional CNN. Another significant issue is the lack of abnormal training samples in the datasets, leading to insufficient training information and the difficulty of designing good classifiers for indicating abnormal events. In addition, there is no chance to train for all possible abnormal events since they can occur unpredictably in real-world environments. Therefore, recent works focus on unsupervised deep learning-based approaches, such as generative approaches [11], [14], [16], to overcome this problem. Finally, the low performance of object localization in pixel-level anomaly detection is also addressed in the literature. Most works achieve high accuracy (measured by Area Under the Curve (AUC)) only on anomaly detection in a frame-level evaluation, while the AUC of object localization in the pixel-level evaluation is much lower. This occurs because of the lack of sufficient features of the objects of interest (e.g. appearance and motion patterns of foreground objects) for model training. These features should be extracted during training in order to learn the model. Specifically, the full input frame is fed into the model without prior knowledge of the objects in the scene, making it difficult for the model to correctly learn the mapping from the appearance to the temporal information of objects and resulting in misdetection and false detection of abnormal objects [10], [11], [14], [16]. Current works try to improve the performance of object localization by isolating patches for deeper feature extraction [7], [13].

Following these considerations, we propose a novel unsupervised spatiotemporal translation based on Generative Adversarial Network (GAN) for anomaly detection and localization in crowded scenes. Our proposed framework, named Deep Spatiotemporal Translation Network (DSTN), is different from the early works [20]–[28], [35] that focus on hand-crafted features since we can handle any possible anomalous event in the complex scenes without tuning parameters during testing, making the proposed DSTN particularly robust, while achieving good running time performance. Additionally, the proposed DSTN is different from [10], [11], [14], [16] that rely on deep learning-based approaches because DSTN is additionally equipped with pre- and post-processing procedures to enhance its detection and localization performance and to eliminate non-object and redundant features. The proposed DSTN has been tested on three challenging anomaly benchmark datasets and compared with other state-of-the-art methods, showing the effectiveness of our proposed framework in terms of both accuracy and time complexity.

To conclude, our main contributions are four-fold:

(i) We propose DSTN, a novel unsupervised deep learning architecture based on GAN, to transform information from the spatial to the temporal domain for addressing the anomaly detection and localization tasks in crowded scenes for surveillance videos. Our DSTN automatically learns the normal samples without varying any parameters, presenting remarkable advantages over previous traditional methods;

(ii) We propose a novel fusion of a background removal frame and a real dense optical flow frame in order to eliminate noise from appearance and motion representations and acquire explicit boundaries of foreground objects;

(iii) We propose concatenated spatiotemporal features to combine important feature information obtained from the new design of patch extraction requiring extensive low-level appearance and motion features;

(iv) This paper presents the first attempt to improve anomaly object localization at the pixel level by introducing an Edge Wrapping technique at the final stage of the framework.

This paper consists of five sections. We review related works in Section 2 and present our proposed method, DSTN, in Section 3. Section 4 shows experimental results compared with several state-of-the-art algorithms and analysis of DSTN. Section 5 provides a conclusion and directions for future work.

II. RELATED WORKS

The related works of video anomaly detection can be grouped into two main categories: traditional-based and deep learning-based approaches.

A. TRADITIONAL-BASED APPROACHES

In this section, we focus on the frameworks that rely on hand-crafted features. These can be divided into two types including temporal (motion) approach and spatiotemporal (appearance and motion) approach. For the temporal approach, X. Tang, *et al.*, [21] proposed abnormal event detection based on motion attention using sparse coding by comparing current regions with neighboring regions to generate a motion attention map. Sparse reconstruction is proposed in [22] by extracting the optical flow and applying the Histogram of Maximal Optical Flow Projections with a sparse representation to generate the dictionary of the normal event. Recently, motion energy [23] and motion entropy [24] were proposed to characterize the abnormal event based on its temporal information only. Overall, the temporal approach is suitable only when dealing with scenes that have a simple background and a low number of foreground objects.

The spatiotemporal approach combines information from both appearance and motion features, making it more robust to complex scenes than the temporal approach. This approach has been addressed by using various local feature descriptors, including Gaussian Mixture Model (GMM) [25], Histogram of Oriented Gradients (HOG) [42], Histogram of Optical Flow (HOF) [42], Histogram of Optical Flow Orientation (HOFO) [43] and Magnitude (HOFM) [26], Gaussian regression [27], and Optical Flow (OF) [35] with Principal Component Analysis (PCA) [60], which can be grouped by applying classifier methods such as K-Means [28] and Bags of Visual Words (BoVW) [27]. However, the problem with the traditional-based approaches is that they rely on hand-crafted features that limit their generalization to other anomalous events.

B. DEEP LEARNING-BASED APPROACHES

Deep learning-based approaches have gained wide popularity as they consistently achieve higher performance than the traditional state-of-the-art approaches [20]–[28], [35] in learning high-level features from a large amount of data and dealing with complex problems such as object detection and recognition and image classification. These approaches can be categorized based on the level of supervision involved. The supervised learning requires labeled data, causing difficulty in detecting unpredictable anomalous events in real-world use cases. Similarly to supervised learning, semi-supervised learning still needs some labeled samples to train the model [15], [29]. In contrast, unsupervised learning is able to handle various anomalous events without any labeling requirement, making it the most suitable approach for anomaly detection in real-world applications. Most frameworks of anomaly detection are based on unsupervised learning because of its high performance in terms of flexibility and reliability of anomaly detection and localization.

Unsupervised learning has been investigated for training in recognition tasks by using CNNs [10], [30]. Ravanbakhsh, *et al.*, [10] proposed a Binary Quantization Layer as a final layer to plug into the top of the network for gathering motion information of abnormality. Xu, *et al.* [7] proposed an Appearance and Motion DeepNet (AMDN) for detecting anomalous events in the videos. The discriminative feature is used instead of hand-crafted features by applying Stacked Denoising AutoEncoders (SDAE) [61]. Fan, *et al.* [13] proposed two-stream variational autoencoder by using Gaussian Mixture Model (GMM) with a Fully Convolutional Network (FCN) [46] at the bottleneck between encoder and decoder to compute the spatial and temporal score. In [17], the authors proposed a neural network for anomaly detection in video surveillance by using three processing blocks; feature learning, sparse representation, and dictionary learning, and also proposed and reformulate an adaptive iterative hard-thresholding algorithm as a new long short-term memory (LSTM).

Liu, *et al.* [16] introduced a video prediction framework for anomaly detection using GANs for training normal events, where the abnormal event is detected by leveraging the difference between a predicted future frame and its ground truth. A future frame is predicted based on appearance and motion feature information. Hasan, *et al.* [15] proposed an end-to-end deep learning framework for abnormal detection using a Convolutional Autoencoder for learning the normal events in crowds and generating the appearance of the normal pattern at testing time, where the abnormality score is measured by the reconstruction error. Similarly to [15], the authors in [14] recently proposed Generative Adversarial Nets (GANs) for a cross-channel abnormal event in which the discriminator is directly used as the final classifiers as an end-to-end anomaly detector. The difference between [15] and [14] is that the latter is based on the interplay between generator and discriminator networks. Another study [11] is dealing with the abnormal event detection in videos using GANs to train only normal events with the use of two networks, (i) generating the optical flow from the frame and (ii) generating the frame from the optical flow.

Following related works, GANs are an outstanding approach that achieves high performance in the anomaly detection task. GANs are a great solution to overcome classification problems as they are able to find the significant features in the frames without any predefined anomaly types. The fundamental architecture of GANs [31]–[33] comprises two networks, the generator G for generating synthetic data z that are likely to come from the same data-generating

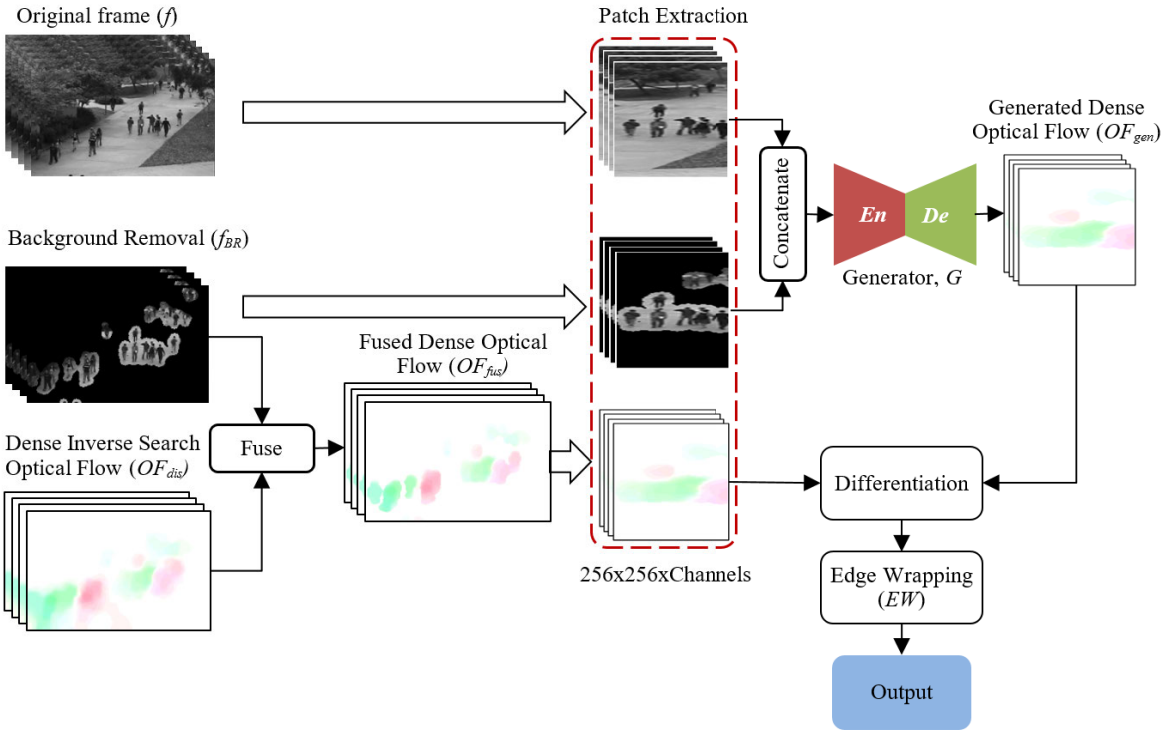


FIGURE 1. The overview of our proposed framework.

distribution as the real samples and discriminator D for discriminating whether the input data are real or fake data generated by G. More specifically, G generates a new image e from random noise z, while D tries to distinguish a real image x from e. In addition, D does its best to classify the synthetic image generated from G as the fake image, while G tries to fool D by producing the synthetic image which looks real, making it challenging to be differentiated. The parameters of G are optimized by updating only with gradients flowing through D in order to maximize the probability of $D(G(z))$ so that D makes a mistake by classifying the synthetic image as the real image, making G efficient in generating images [31]. With enough training time and capacity, G and D are incapable to improve because the probability distributions of the generator and the real data are equal, meaning that D can no longer distinguish between the two distributions. GANs also afford data augmentation and implicit data management due to D, which benefits the deeper training of G on the same small anomaly dataset without training additional classifiers.

Even though GANs outperform several state-of-the-art works, there is still room for improvement of the object localization at the pixel-level evaluation as most of the current works [11], [14], [16] can significantly improve only the performance of frame-level evaluation for the object detection. Thus, apart from the anomaly detection in the frame-level evaluation, our DSTN specifically focuses on improving the performance of anomaly localization at the pixel level. Our model is implemented based on the image-to-image translation framework using the U-Net architecture with skip connections proposed in [34], using the generator with a patch-based discriminator and allowing transforming images to other representations. We take this ability to generate optical flow from raw pixel images by using GANs, so our G is used for spatiotemporal transformation. The difference between our DSTN and [34] is that we use G to learn the normal event to understand its pattern instead of using G to generate a realistic image. At testing time, G is only used for generating appearance (spatial) and motion (temporal) features of the normal event from the input image. With this generated frame, we can simply detect the anomalous areas by comparing the generated frame with the real frame.

III. DSTN FOR ANOMALY DETECTION AND LOCALIZATION

A. OVERVIEW

Our DSTN consists of four main phases including feature collection, spatiotemporal translation, differentiation, and edge wrapping for the object localization. Fig. 1 shows the overview of DSTN that can translate information from the spatial or appearance to the temporal or motion representations.

In the feature collection, we introduce a background removal method, a novel fusion between the background removal frame f_{BR} and the dense inverse search optical flow frame OF_{dis}, a patch extraction, and a concatenation between the original frame f and the background removal frame f_{BR}. Specifically, the novel fusion of f_{BR} and OF_{dis} is proposed to obtain the prior knowledge of the foreground objects in

the scene for the model training. To our knowledge, this is the first attempt to fuse f_{BR} with OF_{dis} to enhance the performance of feature extraction of both appearance and motion patterns for anomaly detection and localization tasks. The background removal method provides the complete shape appearance for each moving foreground object while the dense inverse search optical flow method provides the temporal information corresponding to its input. However, OF_{dis} contains noises that may affect the quality of image generation during GAN training. Thus, due to the performance of the background removal method, we manage to fuse it with OF_{dis} to get rid of noises and make the edges of each foreground object sharper and more precise. The output of this fusion, represented as OF_{fus}, is considered as the real dense optical flow. The fusion of these simple but yet effective techniques provides remarkably good results in noise reduction in OF_{dis} which facilitates G to generate the desired temporal output.

Apart from the fusion, patch extraction is also applied to each frame before input it into a spatiotemporal deep GAN model, consisting of competing G and D networks. Additionally, the concatenation between patches of f and f_{BR} is introduced to capture more information on the moving foreground objects in the scene. This concatenation is specifically designed for delivering the low-level appearance of the moving objects along with their temporal information within the scenes, assisting the model to learn to map the appearance information to temporal information in a more comprehensive way. To conclude, these feature collection methods are introduced in order to obtain better input data to feed into the spatiotemporal deep GAN model. In this way, the model is able to translate the information from the spatial or appearance to the temporal or motion representations efficiently.

In training, G learns only the normal events and translates the spatial to temporal image representations depending on the real dense optical flow. The output of G is a generated dense optical flow, represented as OF_{gen}. In D, it tries to discriminate the patches of real dense optical flow OF_{fus} from the patches of generated dense optical flow OF_{gen} while G tries to fool D by producing more OF_{gen} that is difficult to be discriminated. If D discriminates the patches of OF_{gen} as a fake or wrong image, G will regenerate OF_{gen} until the model reaches the target objective.

In testing, we input all video sequences so that G generates the generated dense optical flow of anomalous events based on the normal events. The anomalous events can be detected by differentiating the pixel intensity of the real optical flow OF_{fus} and the generated dense optical flow OF_{gen}. Finally, we analyze the final output with a novel edge wrapping technique to localize pixels that belong to the anomalous objects. The details of our DSTN are described in the following sections.

B. FEATURE COLLECTION

This is the most important initial task for obtaining the characteristics of objects in the scene. The details of the feature collection approaches are described in the following sections.

1) BACKGROUND REMOVAL

As we consider the real-world situations recorded from the static CCTV cameras, the objects of interest are only the moving foreground objects. In this case, where the background is stationary, we introduce a background removal method, represented as f_{BR}, to extract only the moving foreground object features and to remove unimportant pixels in the background. The f_{BR} image is the representation for appearance information which can be obtained by computing the frame absolute difference between two consecutive frames as shown in (1):

$$f_{BR} = \left| f_t - f_{t-1} \right| \tag{1}$$

where f_t is the current frame and f_{t-1} is the previous frame of a video sequence. In addition, to achieve more appearance features, we implement a binarization on f_{BR} and then concatenate the binarized f_{BR} with f. This concatenation provides more appearance information on the foreground objects of the binarized f_{BR} image, assisting in the learning of G.

In Section 4, we compare the background removal method with a popular technique for background subtraction, i.e., the GMM-based background subtraction [67]. The experimental results clearly show that the background removal method is more effective for anomaly detection in our experiments as it can preserve more appearance information of the moving foreground objects than the GMM-based background subtraction method.

2) FUSED OPTICAL FLOW

Optical flow (OF) is a technique that is used to detect and track the motion of the object of interest obtained from two consecutive frames; f_t and f_{t-1}. Since we consider the motion of foreground objects in terms of running time and accuracy, we choose Dense Inverse Search (DIS), calculated by [37], to generate dense optical flow for our DSTN due to its high performance in real-world applications including low complexity, less time-consumption, and accurate motion detection and tracking. Then we obtain the real dense optical flow generated from the DIS technique, named OF_{dis}. However, OF_{dis} contains some noise dispersed in the scene apart from the objects. Hence, to eliminate it, we propose a novel fusion of f_{BR} and OF_{dis} for appearance and motion, respectively, by integrating these frames to acquire clear foreground object boundaries for the use of object detection, tracking, and localization. Equation (2) shows how to eliminate the noise in DIS optical flow by knowing the information of f_{BR} where its pixel values equal to 0 or 255. Then, the fusion OF_{fus} is defined by applying image masking of f_{BR} on OF_{dis} to change its pixel values. Thus, we obtain the new OF_{dis} represented as OF_{fus} that provides better boundaries of the foreground regions. The output of this fusion OF_{fus} is formulated as below:

$$OF_{fus} = OF_{dis} \left\lfloor \frac{f_{BR}}{f_{BR} + \zeta} \right\rfloor \tag{2}$$

where ζ is a constant value.

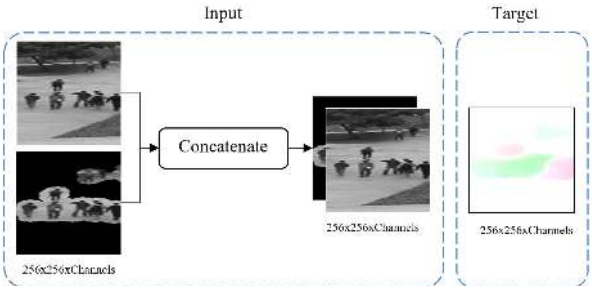


FIGURE 2. The data preparation of concatenated spatiotemporal features for the temporal target output.

3) PATCH EXTRACTION

Patch extraction is important for the feature collection process as it helps to obtain more appearance and motion features. Additionally, the patch extraction allows the model to learn the pattern of local pixels in the scene, resulting in achieving better feature collection performance than extracting the features directly from the full image frame. To extract the patch, we consider the magnitude and direction of the dense optical flow based on the moving objects in the scene. The patch size can be determined by $\frac{w}{a} \times h \times c_p$, where w is the width of the frame, h is the height of the frame, a is a scale value, and c_p represents the number of channels. All patch elements are normalized into a range of $[-1, 1]$. In our DSTN, the patch is extracted by applying a sliding window approach with a stride d to feed into the spatiotemporal translation deep GAN model from its input frames, including f, f_{BR}, and OF_{fus}. While f and f_{BR} are the input for G, OF_{fus} is the input for D. This patch extraction provides the appearance of the moving foreground object along with its motion and direction in the scene, assisting in further processing of the concatenated spatiotemporal features.

4) CONCATENATED SPATIOTEMPORAL FEATURES

In the learning of G, it is important to provide enough information on the appearance to make G understand the features of normal patterns in the scene extensively. The overview of data preparation of concatenated spatiotemporal features is shown in Fig. 2. To achieve more low-level information on the appearance, we propose the concatenation of f and f_{BR} patches for the input of G to learn the normal events. Specifically, the number of channels of the concatenated f and f_{BR} frames is 2 ($c_p = 2$). As a result, the G model obtains efficient information since f_{BR} gives the contour edge information of the foreground objects while f gives the overall information in the scene. After input the concatenated f and f_{BR} frames, the spatiotemporal translation deep GAN will learn this information until it reaches the desired temporal information as the target output.

C. SPATIOTEMPORAL TRANSLATION MODEL

This work investigates the deep spatiotemporal translation GAN network as inspired by the image-to-image translation [34] based on U-Net architecture [63]. The GAN network consists of two cores: generator G and discriminator D, and aims to learn a mapping from the inputs of spatial representation (f and f_{BR}) to the output of temporal representation (OF_{gen}).

1) GENERATOR

The generator G model is the main model of DSTN since it is applied in both training and testing. In the basic GAN [31]–[33], G takes an image x and a random noise z as the input. It generates the output image e with the same resolution as the input x but representing the different channel, using the random noise z, $e = G(x, z)$. In our DSTN, G tries to transform the spatial representation image of the concatenated f and f_{BR} frames to the temporal representation image of generated dense optical flow frame OF_{gen}. However, in this work, the random noise z is not effective to G because the input of G is the spatial representation data and G tries to generate the temporal representation data based on the input data. Hence, this model has been designed to include the drop-out instead of the additional Gaussian noise z. The Drop-Out algorithm [34] is applied within Batch Normalization [62] in the Decoder, resulting e to be reformulated as $e = G(x)$.

Specifically, on the generator architecture, G consists of Encoder (En) and Decoder (De) [34]. Fig. 3 shows the Encoder and Decoder deep network architecture constructed by a residual connection. The Encoder network has been constructed from Convolution (Conv), Batch Normalization (BN), and the Activation Leaky-ReLU (L-ReLU). On the other hand, the Decoder network has been built from De-Convolution (De-Conv), Batch Normalization (BN) with Drop-Out, and the Activation ReLU that allows the model to speed up the learning to suffuse the color space of the training distribution [33]. This residual connection or a skip connection directly connects the encoder layers to the decoder layers based on the architecture of U-Net [63]. The layers of the Encoder and the Decoder are indicated in Fig. 4. In detail, the residual connection is inserted between each layer l at the Encoder and layer t-l at the Decoder, where t is the total number of layers. It allows the information to flow through the initial layer to the last layer by concatenating all channels at layer l with layer t-l. In other words, it often allows one to use smaller networks that are easier to optimize and provide higher quality results of image transformation with a lower complexity cost than the deep convolutional network such as VGG nets [64], [65]. The analysis of the residual connection is discussed in Section 4.

2) DISCRIMINATOR

The discriminator D is used only at the training time. There are two inputs for D to discriminate: the fake patch of OF_{gen} ($OF_{gen} = e$) and the real patch of OF_{fus} ($y = OF_{fus}$) obtained from the fusion between f_{BR} and OF_{dis}. The job of D is to check whether G can produce OF_{gen} or not, and how it looks like comparing with OF_{fus}. D provides a scalar output

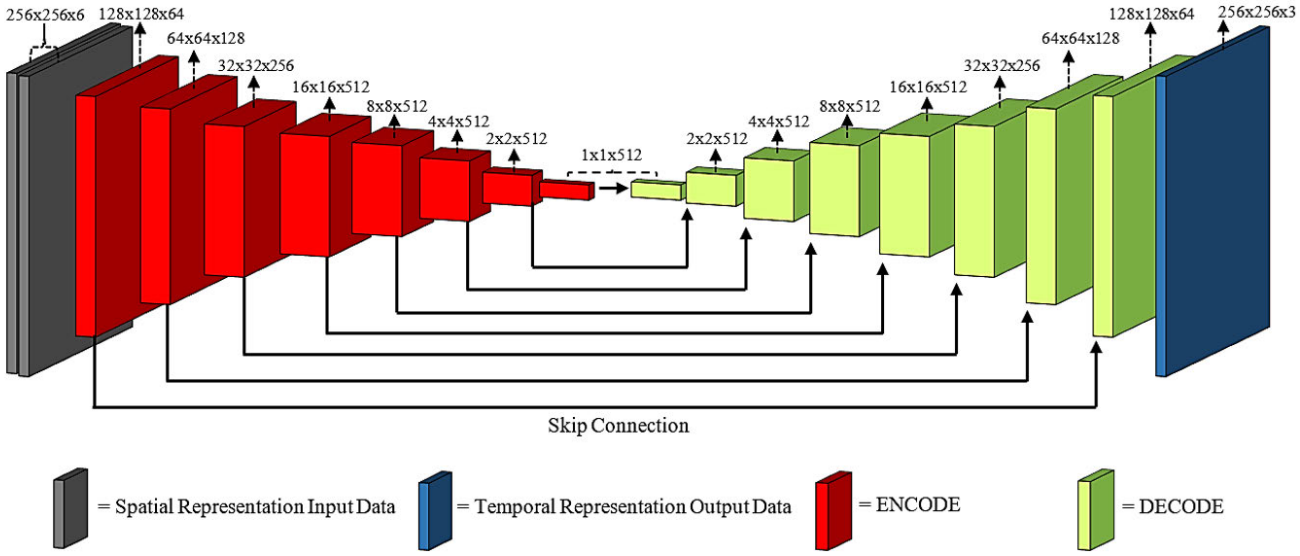


FIGURE 3. The overview of our generator architecture in which its input is a spatial representation and its output is a temporal representation.

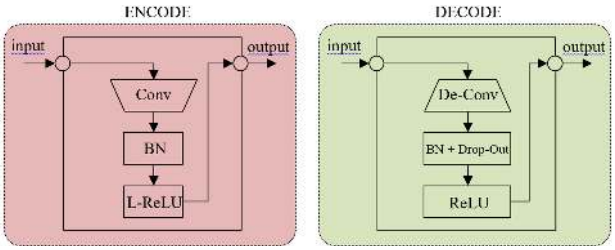


FIGURE 4. The encoder and decoder architecture.

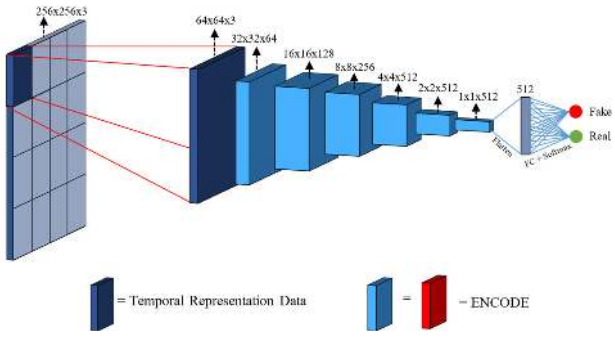


FIGURE 5. The PatchGAN structure in the discriminator architecture.

denoting the probability of the inputs (OF_{fus}, OF_{gen}) for determining the real data.

In D, we use PatchGAN which is constructed as shown in Fig. 5. The PatchGAN can produce a faster training GAN than the full image discriminator net (e.g. 256×256) because it applies to each partial patch of the image. For the implementation of D, the OF_{fus} image is subsampled from the resolution of 256×256 pixels to 64×64 pixels. Hence, the total patches of OF_{fus} image are 16 patches. These 16 patches are passed through the PatchGAN model to decide whether OF_{gen} from G is True or False. We analyze the impact of using 64×64 PatchGAN in Section 4 where we compare the performance of different sizes of PatchGAN in terms of FCN-scores and visual quality outputs.

Two objective functions including a Generator Loss or L1 Loss L_{L1} and a GAN Loss L_{GAN} are determined for training G and D. Our DSTN contains only one network consisting of the translation of spatial to temporal images where the dense optical flow is defined by three-channel components; horizontal, vertical, and magnitude. Suppose y is the target image, which is OF_{fus}, x is the input data of G, which is obtained by concatenating f and f_{BR} frames. Specifically, G learns the mapping from x to y without noise z, where the drop-out algorithm is used in the form of z in this work. The objective functions, L_{L1} and L_{GAN}, can be defined as below,

$$L_{L1}(G) = E_{x,y}\left[\left\|y - G(x)\right\|_1\right], \tag{3}$$

$$L_{GAN}(G, D) = E_y\left[logD(y)\right] + E_x\left[log(1 - D(G(x)))\right]. \tag{4}$$

Finally, the network, G, is optimized as

$$G^* = \arg\min_G \max_D L_{GAN}(G, D) + \lambda L_{L1}(G). \tag{5}$$

This one network of spatiotemporal translation deep GAN provides less complexity cost while contains enough important information for learning normal events. The reason that we do not train for anomalous events is that we need the model to know only normal patterns to be able to handle the possibility of occurrence of various anomalous events without any descriptions for anomaly ground truth samples.

D. ANOMALY DETECTION

After training the normal events by the spatiotemporal learning-based deep GAN, the model understands the translation from the spatial representation of the normal events

(the concatenated frame of f and f_{BR}) to the temporal representation (OF_{fus}). Then, the model parameters of this training are used in the testing procedure.

During testing, all video sequences are used in the experiment. Each frame f and its previous frame $t-1$ from the test video sequences are input into DSTN. We use G in the spatiotemporal learning-based deep GAN as it corresponds to the trained model. In this case, if there are unknown events in the scenes, G will try to generate the dense optical flow based on the normal objects as it has been learned only with the normal events. Thus, it cannot reconstruct the anomalous event in the same way as the normal events. This inaccuracy of G for anomalous event reconstruction leads to the detection of the possible occurrence of anomaly events.

To detect the anomalous events in the scene, we simply subtract the patches of OF_{fus} and OF_{gen} to find the pixel by pixel difference in the scene. In addition, the position of anomalous objects is required to be identified in the scene. Hence, we propose the edge wrapping for object localization in this work. The details of differentiation and edge wrapping are described as following.

1) DIFFERENTIATION

After completing the model training, OF_{gen} can be observed by using the trained model parameters. To identify whether the scene contains the abnormal events or not, the pixel by pixel differentiation between OF_{fus} and OF_{gen} is simply defined by subtracting a patch of OF_{fus} and a patch of OF_{gen} as shown in (6):

$$\Delta_{OF} = OF_{fus} - OF_{gen} > 0 \tag{6}$$

where Δ_{OF} is the subtraction output after differentiating between OF_{fus} and OF_{gen} in which the output value is more than 0. This shows the possible abnormal events in the scene due to the fact that G was not able to reconstruct the anomalous events in OF_{gen} in the same way as the actual anomalous events in OF_{fus}.

After the subtraction, we consider the probability of pixels in Δ_{OF} as the score indicating whether the pixels in Δ_{OF} belong to normal or abnormal events. As each Δ_{OF} from different test video sequences needs to have the same range of pixel values where the lowest value is 0 and the highest value is 1, we consider the highest pixel value in Δ_{OF} as the abnormal pixel in the frame. We normalize Δ_{OF} by computing the maximum value M_{OF} of all components for each test video sequence, regarding its range of values. Then, the ROC curve is computed by gradually changing the threshold of anomaly scores to determine the best decision threshold. The normalization of differentiation Δ_{OF} can be defined as N_{OF} as shown in (7):

$$N_{OF}(i,j) = 1/M_{OF}\,\Delta_{OF}(i,j) \tag{7}$$

where $N_{OF}(i,j)$ is the normalized differentiation of Δ_{OF} in the position of the pixel (i,j).

2) EDGE WRAPPING

After applying differentiation, the differences between OF_{fus} and OF_{gen} are revealed, showing the anomalous events in the scene. However, there are some problems with false anomaly detection on the normal events and over-detection on the abnormal object areas. Thus, to correctly localize the position of the anomalous objects and events in the scene, we propose the Edge Wrapping technique for specifically improving the object localization at the pixel level by preserving only the important edge information and suppressing the rest.

To suppress the unimportant edges along with the noise, we implement the Edge Wrapping based on the Canny edge detection [49]. This Edge Wrapping approach is a multistage procedure divided into three stages, including a noise reduction, a gradient intensity, and a non-maxima suppression. For noise reduction, a Gaussian filter is used to smooth the normalized differentiation output image N_{OF} by removing noise from the background and removing pixels from non-related anomalous events. The size of the filter is $w_e \times h_e \times c_e$ where w_e and h_e represent the width and height of the Gaussian filter of the Edge Wrapping and c_e represents the number of channels such as $c_e = 1$ for the grayscale image and $c_e = 3$ for the color image. For our DSTN, we obtain the grayscale image after differentiation, then $c_e = 1$.

For the gradient intensity, the edge gradient (G_e) can be obtained by convolving the image with a gradient operator in horizontal (G_x) and vertical (G_y) directions. The derivative filter size is the same as the Gaussian filter size in the noise reduction stage. G_e is computed at each pixel using the first derivative to obtain the edge gradient magnitude and the edge gradient direction, which is perpendicular to the edge direction, as shown in (8) and (9) below,

$$G_e = \sqrt{G_x^2 + G_y^2}, \tag{8}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right). \tag{9}$$

Finally, the non-maxima suppression is implemented by determining the threshold to preserve the ridge edges and suppress the noise. We check whether the magnitude at a pixel is greater than a threshold T ($T = 50$). If it is greater than T, there is a point of the edge, representing a local maxima in the neighborhood. Thus, if it is the local maxima, preserve it, otherwise, suppress it to 0. Therefore, we obtain the edges corresponding to the actual anomalous objects. The reason why we choose the threshold value of 50 is indicated in Section 4 where we consider different threshold values in our experiment.

In addition, the Gaussian filter with kernel size $w_e \times h_e \times c_e$ is applied to avoid the occurrence of spot noise in the image. The output of this procedure is represented as EW, which is defined for the final anomaly object localization OL as shown in (10):

$$OL = \Delta_{OF}\left\lfloor \frac{EW}{EW + \zeta} \right\rfloor \tag{10}$$

where ζ is a constant value.

IV. EXPERIMENTAL RESULTS

This section presents the evaluation of our DSTN on three challenging anomaly datasets, including UCSD pedestrian [5], UMN [6], and CUHK Avenue [18], with its implementation details. Our proposed method is analyzed to highlight the impact of residual connections, background removal, patch extraction, and edge wrapping with its base threshold value. The experiment results are comprehensively compared with other state-of-the-art methods in terms of the frame-level and pixel-level evaluations and the time complexity.

A. DATASET

1) UCSD DATASET

The UCSD pedestrian dataset [5] contains crowded scenes in outdoor environments with various anomalous events such as cycling, skateboard, vehicle, and wheelchair. It comprises two sub-sets including Ped1 with 34 training and 16 test video sequences with around 5,500 normal and 3,400 anomalous frames and Ped2 with 16 training and 12 test video sequences with 346 normal and 1,652 anomalous frames. Ped1 has a resolution of 238×158 pixels, while Ped2 has a resolution of 360×240 pixels.

2) UMN DATASET

The UMN dataset [6] has been recorded for distinguishing the anomalous events in crowded scenes. It has 11 video sequences in three different scenes, containing both indoor and outdoor scenes with a total number of 7,700 frames. The image resolution is 320×240 pixels. The main characteristics of this dataset are that the crowds walk normally and then suddenly run into different directions. The walking and running patterns are represented as normal and abnormal events, respectively.

3) CUHK AVENUE DATASET

The CUHK Avenue dataset [18] has been recorded with a fixed camera installed in front of a school gate, containing frames with a total number of 30,652 frames which are divided into 16 training and 21 test video sequences with 15,328 and 15,324 frames, respectively. The length of each video sequence is about 1-2 minutes (around 25 frames per second). The normal pattern includes pedestrians walking parallel to the camera, while the abnormal patterns contain different events (e.g. people throwing objects, jumping, running, and loitering). The ground truth of abnormal object that is labeled in the rectangular area is provided in this dataset.

B. IMPLEMENTATION

The proposed method is implemented by using Python and Matlab based on Keras [50] backend TensorFlow [51]. At training time, we use NVIDIA GeForce GTX 1080 Ti with NVIDIA CUDA Cores 3584 and a memory bandwidth of 484 GB/sec. The testing is implemented by using a 2.8 GHz CPU with Intel Core i9-7960x processor. The reconstruction loss L_{L1} is optimized to 10^{-3} using Adam optimization.

C. EVALUATION CRITERIA

We evaluate the quantitative performance of the proposed DSTN framework based on two criteria: frame level and pixel level. The frame-level evaluation checks whether there is at least one anomalous event that occurs in a test frame, and then the frame is defined as being abnormal. The pixel-level evaluation indicates the position of anomalous events, triggered if the detected abnormal area overlaps more than 40% with the ground truth [20]. The pixel-level evaluation is more challenging than the frame-level evaluation because of the complexity of anomaly localization.

D. EVALUATION ON UCSD DATASET

The first experiment is on the UCSD pedestrian dataset which contains 10 image sequences of the UCSD Ped1 and 12 image sequences of the UCSD Ped2 with the ground truth of pixel-level evaluation. In this dataset, both frame-level and pixel-level protocols are used to evaluate the UCSD Ped1 and the UCSD Ped2.

In the feature collection, we independently extract patches from each original image of the UCSD Ped1 with the size of 238×158 pixels and the UCSD Ped2 with the size of 360×240 pixels to multiple patches with the size of $\frac{w}{4} \times h \times c_p$. The total number of patches of the UCSD Ped1 and the UCSD Ped2 for training is about 22k and 13.6k image patches, respectively. The patches give information on the appearance of the foreground object along with its motion features due to the information of the changing vector within each patch in the frame. After collecting the appearance and motion features, all patches are resized to the resolution of 256×256 pixels to be fed into the model as the input for training and testing.

At the training time, the sizes of the input and target data are set to the resolution of 256×256 pixels as a default. The input of G has been defined by the concatenation of f and f_{BR} patches to provide the information on the appearance with the foreground object boundaries. Since G comprises of Encoder and Decoder networks [34], there are different procedures implemented in each part. In the Encoder network, the image resolution of the first layer of the proposed DSTN framework is 256×256 pixels. Then it is encoded from $256 \rightarrow 128 \rightarrow 64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ to get the variable vectors known as latent space that exploits data in one-dimensional space from the spatial representation of images. The down-scale from the spatial representation image to latent space is implemented by using a CNN with a kernel size 3×3 pixels and a stride of $s = 2$. In addition, the number of neurons in each layer of the Encoder network is set from $6 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512 \rightarrow 512 \rightarrow 512 \rightarrow 512 \rightarrow 512$, corresponding to its image resolution of each input layer.

After the encoding process, the Decoder network starts to generate the target data by performing a reverse process with the same structure. The Decoder decodes the latent space to

TABLE 1. Performance comparison with state-of-the-art methods on UCSD dataset.

Methods	Ped1(frame level)		Ped1(pixel level)		Ped2(frame level)		Ped2(pixel level)	
	EER	AUC	EER	AUC	EER	AUC	EER	AUC
MPPCA [35]	40%	59.0%	81%	20.5%	30%	69.3%	-	-
Social force (SF) [6]	31%	67.5%	79%	19.7%	42%	55.6%	80%	-
SF+MPPCA [5]	32%	68.8%	71%	21.3%	36%	61.3%	72%	-
Sparse Reconstruction [19]	19%	-	54%	45.3%	-	-	-	-
MDT [5]	25%	81.8%	58%	44.1%	25%	82.9%	54%	-
Detection at 150fps [18]	15%	91.8%	43%	63.8%	-	-	-	-
SR+VAE [8]	16%	90.2%	41.6%	64.1%	18%	89.1%	-	-
AMDN (Double Fusion) [7]	16%	92.1%	40.1%	67.2%	17%	90.8%	-	-
GMM [4]	15.1%	92.5%	35.1%	69.9%	-	-	-	-
Plug-and-Play CNN [10]	8%	95.7%	40.8%	64.5%	18%	88.4%	-	-
GANs [11]	8%	97.4%	35%	70.3%	14%	93.5%	-	-
GMM-FCN [13]	11.3%	94.9%	36.3%	71.4%	12.6%	92.2%	**19.2%**	78.2%
Convolutional AE [15]	27.9%	81%	-	-	21.7%	90%	-	-
Liu et al [16]	23.5%	83.1%	-	33.4%	12%	95.4%	-	40.6%
Adversarial Discriminator [14]	7%	96.8%	34%	70.8%	11%	**95.5%**	-	-
AnomalyNet [17]	25.2%	83.5%	-	45.2%	10.3%	94.9%	-	52.8%
DSTN (Proposed Methed)	**5.2%**	**98.5%**	**27.3%**	**77.4%**	**9.4%**	**95.5%**	21.8%	**83.1%**

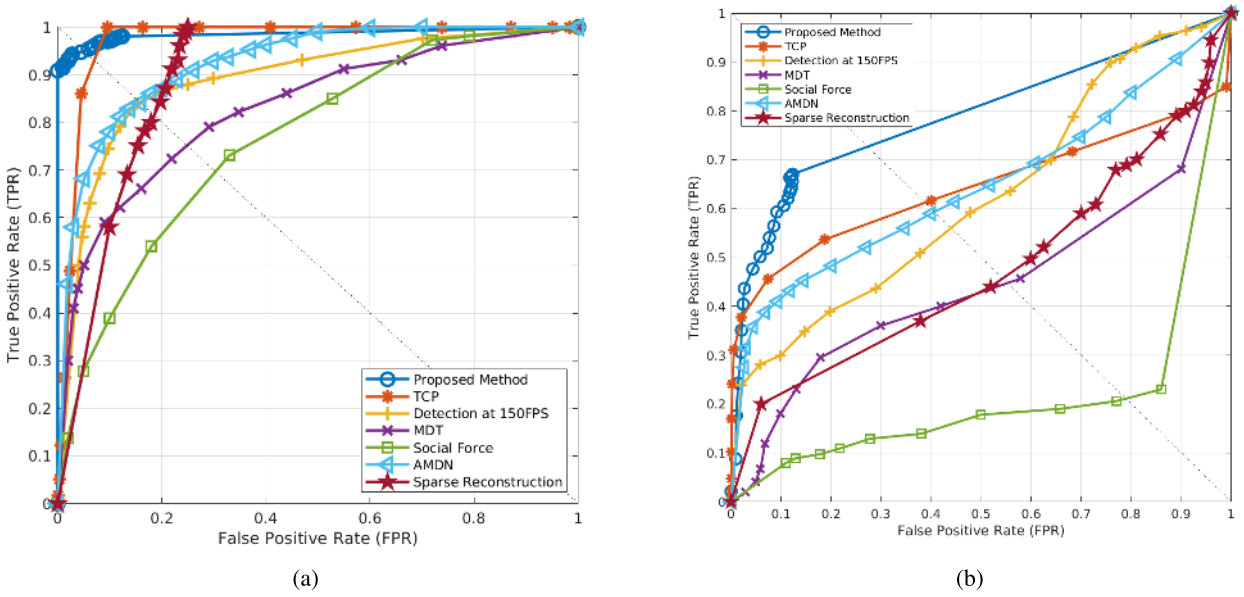


FIGURE 6. ROC comparison on the UCSD Ped1 dataset: (a) frame-level evaluation and (b) pixel-level evaluation.

the target image size of 256×256 pixels in order to reach the temporal representation of the optical flow output. The number of neurons in each layer of the Decoder is the same as the Encoder configuration with its image resolution of the input layer. Moreover, the drop-out is applied in the Decoder to be represented as the random noise z of GAN by removing connections of neurons with the default probability at $p = 0.5$. This drop-out helps to prevent over-fitting on the training dataset.

Furthermore, the training process requires D to vary G in order to optimize the distinction of a fake and a real image. D is represented by PatchGAN, having as input size 64×64 pixels and output the probability showing whether the object belongs to a negative class (fake) or a positive class (real). The PatchGAN structure is defined as $64 \rightarrow 32 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$, where it is flattened to 512 neurons which are then followed by a Fully Connection (FC) and a Softmax layer to link to the target output label. Since PatchGAN works on a partial image which has less learnable parameters, we observe that the training of the deep spatiotemporal translation GAN network is faster. For other parameter settings, the batch size is set to 1 and the reconstruction loss (norm L1) is optimized to be lower than 0.001. Adam optimization is used with a learning rate of 0.0002 and a momentum of 0.9.

At the testing time, G is the only model used to generate OF_{gen} to compare with the original temporal representation OF_{fus}. The resolution of the test images is the same as the training images for all datasets. Various state-of-the-art methods [4]–[8], [10], [11], [13]–[19], [35], [36] are compared with our DSTN. According to the quantitative comparison of different methods in terms of Equal Error Rate (EER) and Area Under Curve (AUC) in Table 1, it is clearly shown that our DSTN outperforms all the methods as we achieve

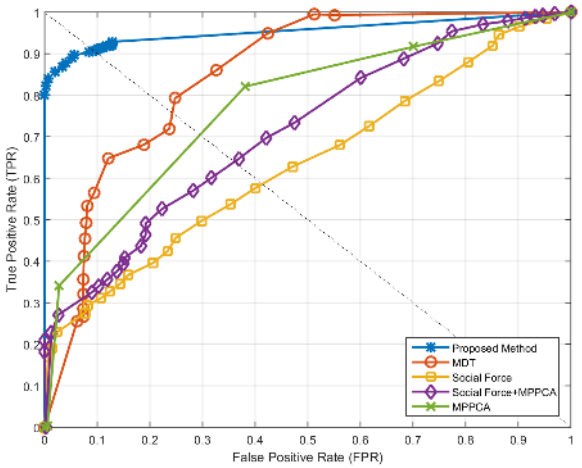


FIGURE 7. ROC comparison on the UCSD Ped2 dataset at the frame level.

the highest AUC value in both frame-level and pixel-level evaluations of the UCSD pedestrian dataset. We also reach the lowest EER value compared to the other methods except only for the pixel-level evaluation on the UCSD Ped2 in [13].

The qualitative results of our proposed method can be visually illustrated in the standard protocol for abnormality detection as ROC curves, where the x-axis is the False Positive Rate (FPR) and the y-axis is the True Positive Rate (TPR). To produce the ROC curves, the threshold parameter has been varied from 0 to 1 to indicate the flow of TPR and FPR. We compare our performance with other state-of-the-art methods from their original papers (when available) as shown in Fig. 6 and Fig. 7, where Fig. 6 shows the ROC comparison on the UCSD Ped1 in both (a) frame-level evaluation and (b) pixel-level evaluation and Fig. 7 shows the ROC comparison on the UCSD Ped2 in the frame-level evaluation. According to Fig. 6 and Fig. 7, our proposed DSTN, represented as the dark blue curves, outperforms all the competing methods as our curves have the strongest growth on the TPR, meaning that the abnormal events in our proposed method are accurately detected and localized in both frame-level and pixel-level evaluations. We also show some examples of the anomaly detection and localization on the UCSD dataset in Fig. 8. The results show that our DSTN can efficiently detect different anomalous events in the frame including a single object (e.g. a wheelchair, a vehicle, a skateboard, and a bicycle) and multiple objects (e.g. bicycles, vehicle and bicycle, bicycle and skateboard). However, there is a false anomaly detection in Fig. 8 (h), where the proposed method detects the normal event (walking pedestrians represented in red color) as an abnormal event. This is probably because the speed of walking pedestrians is the same as the cycling event in the scene.

E. EVALUATION ON UMN DATASET

We evaluate the performance on the UMN dataset using the same training parameter settings and network configuration as for the UCSD dataset. Table 2 shows the AUC comparison of our DSTN performance with other state-of-the-art methods [6], [10], [11], [14], [17], [19], [36]. From Table 2, it is clear that the proposed DSTN outperforms most of the baseline methods and its AUC performance is equal to the best method [17]. The examples of anomaly detection and localization on three different scenes of the UMN dataset are shown in Fig. 9.

TABLE 2. AUC comparison with state-of-the-art methods on UMN dataset.

Method	AUC
Optical-flow [6]	0.84
SFM [6]	0.96
Sparse Reconstruction [19]	0.976
Commotion [36]	0.988
Plug-and-Play CNN [10]	0.988
GANs [11]	0.99
Adversarial Discriminator [14]	0.99
AnomalyNet [17]	**0.996**
DSTN (Proposed Method)	**0.996**

F. EVALUATION ON CUHK AVENUE DATASET

In this section, we follow the previous training parameter settings and network configuration of the UCSD and UMN datasets for the evaluation on the CUHK Avenue dataset. Table 3 shows the comparison of our DSTN performance with other state-of-the-art methods [13], [15]–[18], in which the proposed DSTN outperforms all the competing methods for both AUC and EER. Fig. 10 presents examples of anomaly detection and localization on the CUHK Avenue dataset, containing multiple abnormal activities including (a) jumping, (b) throwing objects (papers), (c) falling objects (papers), and (d) grabbing a falling bag. From Fig. 10, it is clearly seen that our DSTN can detect and localize various anomalous events accurately, especially in Fig. 10 (d) where the abnormal areas (e.g. a bag and a human head) are detected, even though they have only a slight difference in motion from the normal events.

G. ANALYSIS OF RESIDUAL CONNECTION

As the residual connection or the skip connection in G is significant to our DSTN, we conduct additional experiments to indicate and analyze the performance of the residual connection compared to the autoencoder network which is created by removing the residual connections in the U-Net. First, we train on all training video sequences from the UCSD Ped2 dataset for 40 epochs on both networks to study their performance of minimizing the L1 loss on the training samples as shown in Fig. 11. The residual connection loss, represented as a red star curve, exhibits lower training error over training time compared to the autoencoder loss represented as a blue dash curve, meaning that the performance of the residual connection is remarkably higher than the one of the autoencoder.

In addition, we observe the ability of temporal information generation of the residual connection and the autoencoder from the test video sequences of the UCSD Ped2 dataset as

(a) wheelchair (b) vehicle (c) skateboard (d) bicycle

(e) bicycles (f) vehicle and bicycle (g) bicycle and skateboard (h) bicycle and skateboard

FIGURE 8. Examples of anomaly detection and localization results on the UCSD Ped1 and Ped2 dataset.

(a) running in an outdoor scene (b) running in an outdoor scene (c) running in an indoor scene (d) running in an outdoor scene

FIGURE 9. Examples of anomaly detection and localization results on the UMN dataset.

TABLE 3. Performance comparison with state-of-the-art methods on CUHK Avenue dataset.

Method	EER	AUC
Convolutional AE [15]	25.1%	70.2%
Detection at 150 FPS [18]	-	80.9%
GMM-FCN [13]	22.7%	83.4%
Liu et al [16]	-	85.1%
AnomalyNet [17]	22%	86.1%
DSTN (Proposed Method)	**20.2%**	**87.9%**

shown in Fig. 12. Fig. 12 (c) shows that the autoencoder is unable to generate dense optical flow in our experiment. On the other hand, the residual connection in Fig. 12 (b) can properly generate new dense optical flow corresponding to the real dense optical flow in Fig. 12 (a), providing a good quality result of the synthesized image.

Besides the above, we also compute FCN-scores on pixel accuracy [59] and Structural SIMilarity Index (SSIM) [66] metrics on the UCSD Ped2 dataset to compare the performance between the autoencoder and the residual connection as shown in Table 4. For both evaluations, a higher value means a better result. The pixel accuracy metric is a common semantic segmentation evaluation. In this work, there are two classes; a foreground region class and a background region class. Let n_{ij} be the number of wrong classified pixels of

(a) jumping (b) throwing objects (c) falling objects (d) grabbing an object

FIGURE 10. Examples of anomaly detection and localization results on the CUHK Avenue dataset.

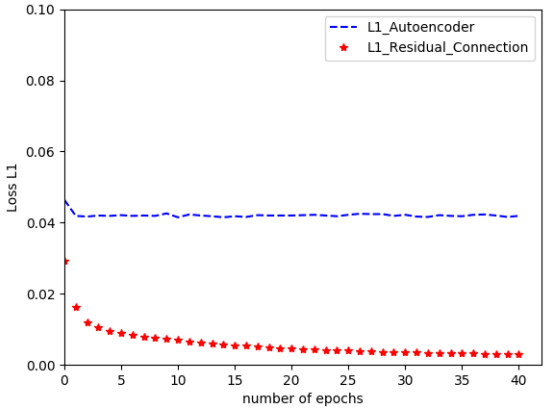


FIGURE 11. Performance comparison between the autoencoder and the residual connection on the UCSD Ped2 dataset.

(a) dense optical flow (b) residual connection (c) autoencoder

FIGURE 12. Examples of dense optical flow generation results of residual connection and autoencoder on the UCSD Ped2 dataset.

TABLE 4. Performance comparison of the autoencoder and the residual connection in terms of FCN-scores on pixel accuracy and Structural SIMilarity Index (SSIM) on the UCSD Ped2 dataset.

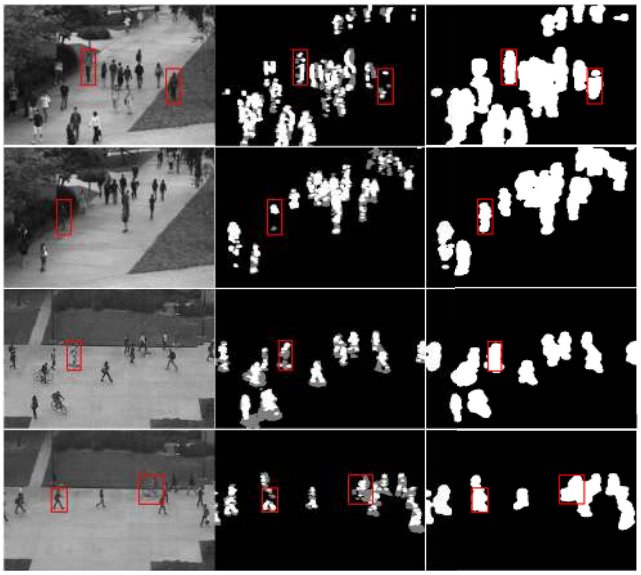
Network Architecture	Pixel Accuracy	SSIM
autoencoder	0.83	0.82
residual connection	**0.9**	**0.96**

class i, and n_{ti} be the total number of pixels of class i. The pixel accuracy can be computed by $\sum_i n_{ii} / \sum_i n_{ti}$. For the SSIM index, we use it to measure the similarity between the original and the synthesized images. The more the synthesized image looks like the original image, the more efficient the model is. The results in Table 4 show that the residual connection clearly achieves superior results on the low-level information than the autoencoder for both pixel accuracy and SSIM evaluations.

H. ANALYSIS OF DSTN

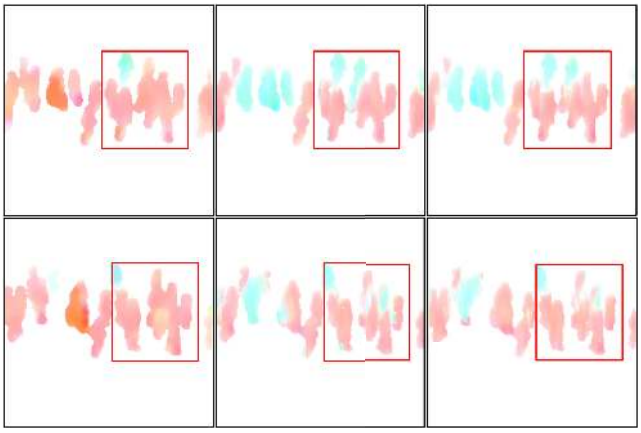
In this section, the proposed DSTN is analyzed to emphasize the significance of its main element. First of all, to demonstrate the performance of the background removal method using the frame absolute difference on the proposed DSTN, we compare it with a popular technique for background subtraction, i.e., the Gaussian mixture model (GMM)-based background subtraction method [67], on the UCSD dataset as shown in Fig. 13. As we train only the normal event patterns in the scene, Fig. 13 (c) shows that the background removal method can preserve more information on the normal events than the GMM-based background subtraction method which loses some appearance information of the normal and abnormal events as shown in the red box in Fig. 13 (b), providing incomplete and inaccurate information of the foreground objects. According to these experimental results, the background removal method is more suitable for our DSTN since it comprehensively preserves the appearance feature information of the moving foreground objects. Thus, we use it as the foreground feature extractor under the assumptions of static CCTV cameras.

Considering the impact of using the patch in our proposed method, we investigate different sizes of PatchGAN used in D to demonstrate its performance to the DSTN. Based on [34], the full ImageGAN has greater depth and more parameters than PatchGAN, making it more difficult to train. Thus, we test additional PatchGAN with a patch size of 32×32 pixels and 64×64 pixels. The use of the 32×32 PatchGAN provides lower intensity on the appearance of objects than the 64×64 PatchGAN which is better in the visual quality of the synthesized images, meaning that the structure of synthesized images is more recognizable, as shown in Fig. 14. We also compute the FCN-scores on the



(a) original frame (b) GMM (c) background removal

FIGURE 13. Performance comparison of background subtraction between (b) GMM-based background subtraction method and (c) background removal method on the UCSD dataset.



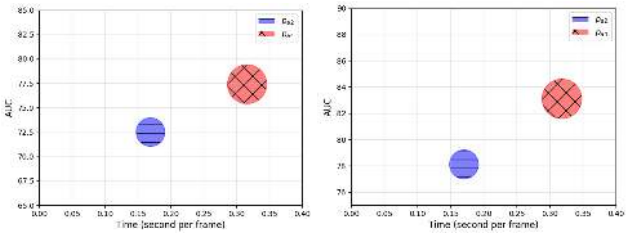
(a) frame (b) 32×32 pixels (c) 64×64 pixels

FIGURE 14. Comparison of different sizes of PatchGAN: (b) 32 × 32 pixels and (c) 64 × 64 pixels.

TABLE 5. Performance comparison of different sizes of PatchGAN in terms of FCN-scores on pixel accuracy and Structural SIMilarity Index (SSIM) on the UCSD Ped2 dataset.

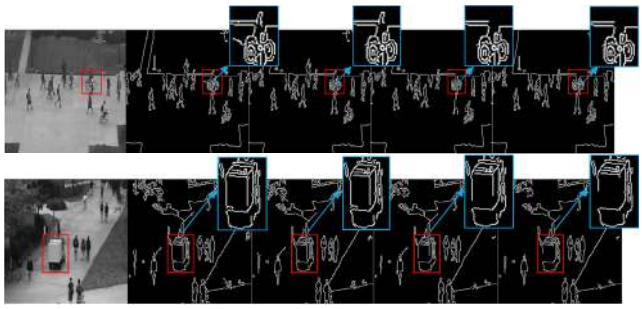
PatchGAN size	Pixel Accuracy	SSIM
32×32	0.89	**0.96**
64×64	**0.9**	**0.96**

pixel accuracy and the SSIM of the 32×32 PatchGAN and the 64×64 PatchGAN as shown in Table 5. From Table 5, the 64×64 PatchGAN achieves slightly better pixel accuracy than the 32×32 PatchGAN. Thus, according to the performance of the 64×64 PatchGAN in Fig. 14 and Table 5, we decided to use it in all the experiments.



(a) Ped1 (b) Ped2

FIGURE 15. Comparison of AUC and computational complexity of two different patch sizes, p_{a2} and p_{a4}, on the UCSD datasets.



(a) frame (b) T=35 (c) T=50 (d) T=65 (e) T=80

FIGURE 16. Comparison of edge detection with different thresholds: 35, 50, 65, and 80.

Furthermore, we also ran additional experiments to show the effect of the patch extraction from the feature collection process. We investigate two different patch sizes with the scale value a equal to 2 (p_{a2}) and a equal to 4 (p_{a4}) on the UCSD datasets. Fig. 15 shows the comparison of AUC and computational complexity of two different patch sizes, p_{a2} and p_{a4}, on the UCSD datasets. p_{a2} provides low computational complexity as it achieves 50% faster processing than p_{a4} due to its bigger patch size. However, p_{a2} has a lower accuracy than p_{a4} on both frame-level and pixel-level evaluations. Specifically, the AUC values of p_{a2} on the UCSD Ped1 dataset are 96.9% for frame level and 72.5% for pixel level, while the AUC values of p_{a4} are 98.5% for frame level and 77.4% for pixel level. For the AUC values of p_{a2} on the UCSD Ped2 dataset, they are 95.4% for frame level and 78.1% for pixel level, while the AUC values of p_{a4} are 95.5% for frame level and 83.1% for pixel level. This remarkably shows that p_{a4} achieves more accurate results for both evaluations. Based on these experimental results, we can conclude that the patch size with higher scale value provides better abnormal event localization. Since we aim to collect features from both appearance and motion information for enhancing the localization accuracy, we use p_{a4} for the training videos of all datasets. The stride d is assigned to $\frac{w}{a}$ for extracting the patches which are then resized to 256×256 pixels. Thus, the patch size is $\frac{w}{4} * h * c_p$.

As we aim to improve the performance of the anomaly localization in the pixel-level evaluation, we introduce the

TABLE 6. Impact of Edge Wrapping (*EW*) on UCSD frame-level and pixel-level performances.

Methods	Ped1(frame level)		Ped1(pixel level)		Ped2(frame level)	
	EER	AUC	EER	AUC	EER	AUC
DSTN without EW	9%	95.8%	35.6%	70.1%	9.8%	94.6%
DSTN with EW	**5.2%**	**98.5%**	**27.3%**	**77.4%**	**9.37%**	**95.54%**

TABLE 7. Computational time comparison during testing (seconds per frame).

Methods	CPU	GPU	Memory	Running Time			
				Ped1	Ped2	UMN	Avenue
Sparse Reconstruction [19]	2.6GHz	-	2.0GB	3.8	-	0.8	-
Detection at 150 fps [18]	3.4GHz	-	8.0GB	**0.007**	-	-	**0.007**
MDT [5]	3.9GHz	-	2.0GB	17	23	-	-
Li et al [20]	2.8GHz	-	2.0GB	0.65	0.80	-	-
AMDN (Double Fusion) [7]	2.1GHz	Nvidia Quadro K4000	32GB	5.2	-	-	-
DSTN (Proposed Method)	2.8GHz	-	24GB	0.315	**0.319**	**0.318**	0.334

(a) DSTN without Edge Wrapping

(b) DSTN with Edge Wrapping

FIGURE 17. Examples of the impact of Edge Wrapping on all datasets: UCSD Ped1, UCSD Ped2, UMN, and CUHK Avenue.

Edge Wrapping (EW) at the final stage of our DSTN. To choose the threshold values in EW, Canny edge detection [49] recommends the ratio of the high to the low threshold in the range two or three to one. In this work, the low threshold is observed from the high threshold divided by three. Since the pixels above the high threshold value considered as strong edges have the maximum value of 255, the lower threshold should be assigned as $\frac{255}{3} = 85$. Then, we explore different threshold values, including the threshold value of 35, 50, 65, and 80. We conduct experiments on edge preservation of different threshold values as shown in Fig. 16. The experimental results show that the threshold value of 50 ($T = 50$) can preserve better edges than other threshold values. Specifically, the threshold values of 35 ($T = 35$) and 50 ($T = 50$) are better than other threshold values ($T = 65$, $T = 80$) because they can preserve more soft edges of the objects in the scene, while the threshold values of 65 and 80 give incomplete edge results. However, the threshold value of 35 provides more edges (e.g. object shadows and background) which are not useful in our experiment. Thus, in this work, we select the threshold value of 50 as the base threshold.

Table 6 shows a comparison of the impact of *EW* on the DSTN for the frame-level and pixel-level performances on the UCSD dataset. Using *EW*, we achieve a significant improvement in terms of the AUC and EER, especially in the pixel-level localization. To further demonstrate the importance of *EW*, we show a comparison of applying *EW* on examples from all datasets, the UCSD, UMN, and CUHK Avenue, in Fig. 17. From Fig. 17, it is clear that *EW* helps to locate the actual anomalous objects more precisely since all unrelated features (e.g. shadows, noises, and normal objects) are suppressed. These results prove the benefit of applying *EW* for anomaly detection and localization in combination with the proposed DSTN.

I. ANALYSIS ON TIME COMPLEXITY

We compare the computational time of the proposed DSTN with other state-of-the-art methods [5], [7], [18]–[20]. As these methods do not provide their original implementations, we follow the computational time and the environment from [7]. With regard to computational time in frame per second (fps), our DSTN achieves 3.17 fps, 3.15 fps, 3.15 fps, and 3 fps on the UCSD Ped1, UCSD Ped2, UMN, and CUHK Avenue datasets, respectively. We also compare our time complexity in seconds per frame with other baseline methods as shown in Table 7. It is clear that our computational time is lower than for most of the baseline methods except for [18]. This is because our architecture is based on a deep learning framework consisting of multiple convolutional layers while [18] is based on a sparse combination learning framework that has lower neuron connections. However, we obtain significantly higher AUC value and relatively much lower EER value in both frame-level and pixel-level evaluations on the UCSD and the CUHK Avenue datasets than [18]. According to our experimental results, we can conclude that the proposed DSTN outperforms other competing methods by achieving the highest AUC value in both frame-level and pixel-level evaluations while providing a good running time for surveillance videos.

V. CONCLUSION

In this paper, we propose a novel unsupervised spatiotemporal anomaly detection and localization for surveillance videos. The proposed DSTN framework is embedded with concepts of deep convolution neural network of GAN based Edge Wrapping approach which brings advantages to anomaly localization. The deep spatiotemporal translation network is designed to learn the appearance and motion representations with the use of the fusion and the concatenation of patches for combining the learned features. Additionally, our proposed method does not rely on any prior knowledge in order to design features for the input (as we use raw pixels) and does not involve low-level object analysis, such as object detection and tracking. We provide extensive experimental results compared with other state-of-the-art methods and implemented on three publicly available datasets including the UCSD pedestrian, UMN, and CUHK Avenue. We clearly show that our DSTN outperforms other state-of-the-art methods in terms of accuracy and time complexity as we obtain the highest AUC value in both frame-level and pixel-level evaluations for all datasets and achieve good running time that outperforms most of the baseline methods. Our method is effective and robust for anomaly event detection and localization in the crowded scenes for surveillance videos. For future work, we will explore an object translation model with a clustering method to enhance the performance of the anomaly detection and localization from the complex scene. Other abnormalities will be observed for increasing the robustness of the model for real-world use.

REFERENCES

[1] *Video Surveillance Intelligence Service—Annual—IHS Technology*. Accessed: Aug. 5, 2019. [Online]. Available: https://technology.ihs.com/Services/570988/video-surveillance-intelligence-service-annual

[2] T. Akinbinu and Y. Mashalla, "Impact of computer technology on health: Computer Vision Syndrome (CVS)," *Med. Pract. Rev.*, vol. 5, no. 3, pp. 20–30, 2014.

[3] K. Gates, "Professionalizing police media work: Surveillance video and the forensic sensibility," in *Images, Ethics, Technology*. Evanston, IL, USA: Routledge, 2015, pp. 53–69.

[4] Y. Feng, Y. Yuan, and X. Lu, "Learning deep event models for crowd anomaly detection," *Neurocomputing*, vol. 219, pp. 548–556, Jan. 2017.

[5] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1975–1981.

[6] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 935–942.

[7] D. Xu, Y. Yan, E. Ricci, and N. Sebe, "Detecting anomalous events in videos by learning deep representations of appearance and motion," *Comput. Vis. Image Understand.*, vol. 156, pp. 117–127, Mar. 2017.

[8] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33353–33361, 2018.

[9] S. Bouindour, M. M. Hittawe, S. Mahfouz, and H. Snoussi, "Abnormal event detection using convolutional neural networks and 1-class SVM classifier," in *Proc. 8th Int. Conf. Imag. Crime Detection Prevention (ICDP)*, 2017, pp. 1–6.

[10] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, and N. Sebe, "Plug-and-play CNN for crowd motion analysis: An application in abnormal event detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 1689–1698.

[11] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1577–1581.

[12] H. Wei, Y. Xiao, R. Li, and X. Liu, "Crowd abnormal detection using two-stream fully convolutional neural networks," in *Proc. 10th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA)*, Feb. 2018, pp. 332–336.

[13] Y. Fan, G. Wen, D. Li, S. Qiu, and M. D. Levine, "Video anomaly detection and localization via Gaussian mixture fully convolutional variational autoencoder," 2018, *arXiv:1805.11223*. [Online]. Available: http://arxiv.org/abs/1805.11223

[14] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1896–1904.

[15] 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.

[16] 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.

[17] 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 Secur.*, vol. 14, no. 10, pp. 2537–2550, Oct. 2019.

[18] 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.
[19] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3449–3456.
[20] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.
[21] X. Tang, S. Zhang, and H. Yao, "Sparse coding based motion attention for abnormal event detection," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 3602–3606.
[22] A. Li, Z. Miao, Y. Cen, and Q. Liang, "Abnormal event detection based on sparse reconstruction in crowded scenes," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1786–1790.
[23] T. Chen, C. Hou, Z. Wang, and H. Chen, "Anomaly detection in crowded scenes using motion energy model," *Multimedia Tools Appl.*, vol. 77, no. 11, pp. 14137–14152, Jun. 2018.
[24] Z. Wang, C. Hou, B. Li, T. Chen, L. Yao, and M. Song, "Global abnormal event detection in video via motion information entropy," in *Proc. 2nd URSI Atlantic Radio Sci. Meeting (AT-RASC)*, May 2018, pp. 1–4.
[25] D. Du, H. Qi, Q. Huang, W. Zeng, and C. Zhang, "Abnormal event detection in crowded scenes based on structural multi-scale motion interrelated patterns," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.
[26] R. V. H. M. Colque, C. Caetano, M. T. L. de Andrade, and W. R. Schwartz, "Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 27, no. 3, pp. 673–682, Mar. 2017.
[27] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2909–2917.
[28] S. Cosar, G. Donatiello, V. Bogorny, C. Garate, L. O. Alvares, and F. Bremond, "Toward abnormal trajectory and event detection in video surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 683–695, Mar. 2017.
[29] Y. Zhang, H. Lu, L. Zhang, and X. Ruan, "Combining motion and appearance cues for anomaly detection," *Pattern Recognit.*, vol. 51, pp. 443–452, Mar. 2016.
[30] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," 2017, *arXiv:1710.09282*. [Online]. Available: http://arxiv.org/abs/1710.09282
[31] 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.*, 2014, pp. 2672–2680.
[32] Tim Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.
[33] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434
[34] 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. 1125–1134.
[35] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2921–2928.
[36] H. Mousavi, M. Nabi, H. Kiani, A. Perina, and V. Murino, "Crowd motion monitoring using tracklet-based commotion measure," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 2354–2358.
[37] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 471–488.
[38] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 17–31, Apr. 2007.
[39] M. J. Roshtkhari and M. D. Levine, "An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions," *Comput. Vis. Image Understand.*, vol. 117, no. 10, pp. 1436–1452, Oct. 2013.
[40] T. Xiao, C. Zhang, and H. Zha, "Learning to detect anomalies in surveillance video," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1477–1481, Sep. 2015.
[41] Y. Cong, J. Yuan, and Y. Tang, "Video anomaly search in crowded scenes via spatio-temporal motion context," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1590–1599, Oct. 2013.
[42] Y. Yuan, Y. Feng, and X. Lu, "Statistical hypothesis detector for abnormal event detection in crowded scenes," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3597–3608, Nov. 2017.
[43] N. Patil and P. K. Biswas, "Global abnormal events detection in crowded scenes using context location and motion-rich spatio-temporal volumes," *IET Image Process.*, vol. 12, no. 4, pp. 596–604, Apr. 2018.
[44] M. Sabokrou, M. Fayyaz, M. Fathy, and R. Klette, "Deep-cascade: Cascading 3D deep neural networks for fast anomaly detection and localization in crowded scenes," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1992–2004, Apr. 2017.
[45] Y. Yuan, Y. Feng, and X. Lu, "Structured dictionary learning for abnormal event detection in crowded scenes," *Pattern Recognit.*, vol. 73, pp. 99–110, Jan. 2018.
[46] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Comput. Vis. Image Understand.*, vol. 172, pp. 88–97, Jul. 2018.
[47] S. Wang, E. Zhu, J. Yin, and F. Porikli, "Video anomaly detection and localization by local motion based joint video representation and OCELM," *Neurocomputing*, vol. 277, pp. 161–175, Feb. 2018.
[48] X. Zhang, S. Yang, X. Zhang, W. Zhang, and J. Zhang, "Anomaly detection and localization in crowded scenes by motion-field shape description and similarity-based statistical learning," 2018, *arXiv:1805.10620*. [Online]. Available: http://arxiv.org/abs/1805.10620
[49] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.
[50] Keras-Team. (Nov. 6, 2019). *Keras*. GitHub. Accessed: Nov. 12, 2019. [Online]. Available: https://github.com/keras-team/keras
[51] M. Abadi, "TensorFlow: A system for large-scale machine learning," in *Proc. Symp. Operating Syst. Design Implement.*, 2016, pp. 265–283.
[52] M. Sabokrou, M. Fathy, and M. Hoseini, "Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder," *Electron. Lett.*, vol. 52, no. 13, pp. 1122–1124, Jun. 2016.
[53] Z. Fang, F. Fei, Y. Fang, C. Lee, N. Xiong, L. Shu, and S. Chen, "Abnormal event detection in crowded scenes based on deep learning," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14617–14639, Nov. 2016.
[54] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 2015, pp. 1135–1143.
[55] K. Ullrich, E. Meeds, and M. Welling, "Soft weight-sharing for neural network compression," 2017, *arXiv:1702.04008*. [Online]. Available: http://arxiv.org/abs/1702.04008
[56] T. N. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, and B. Ramabhadran, "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6655–6659.
[57] P. Maji and R. Mullins, "On the reduction of computational complexity of deep convolutional neural networks," *Entropy*, vol. 20, no. 4, p. 305, 2018.
[58] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[59] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[60] E. L. Andrade, S. Blunsden, and R. B. Fisher, "Modelling crowd scenes for event detection," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, 2006, pp. 175–178.
[61] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[62] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167
[63] 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.
[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556
[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[66] 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.
[67] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th IEEE Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2004, pp. 28–31.

THITTAPORN GANOKRATANAA (Graduate Student Member, IEEE) received the B.Sc. degree (Hons.) in media technology from the King Mongkut's University of Technology Thonburi, Bangkok, Thailand, in 2015, and the M.Eng. degree in electrical engineering from Chulalongkorn University, Bangkok, in 2017, where she is currently pursuing the Ph.D. degree in electrical engineering. In 2018, she was a Lecturer in image processing at King Mongkut's University of Technology Thonburi. Her research interests include computer vision and machine learning, specifically human–computer interaction for surveillance videos. She is serving as the President of the IEEE HKN Mu Theta Chapter. She received the Royal Monogram PPR Brooch Investiture from the Majesty King Phra Pok Klao (King Rama VII) and the Majesty Queen Rambai Barni of Siam at the Army Artillery Club, Pol. AAA, (Antiaircraft Artillery), the best paper awards from ICACME and SKIMA, the best student paper awards SKIMA and SNLP, the medals from the 47th International Exhibition of Inventions of Geneva, the 1st Runner-up of Three Minute Thesis competition, and the IEEE-HKN 2015-2016 Outstanding Chapter Awards Winner.

SUPAVADEE ARAMVITH (Senior Member, IEEE) received the B.S. degree (Hons.) in computer science from Mahidol University, in 1993, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, USA, in 1996 and 2001, respectively. In June 2001, she joined Chulalongkorn University, where she is currently an Associate Professor at the Department of Electrical Engineering, with a specialization in video technology. She has successfully advised 32 bachelor's, 27 master's, and nine Ph.D. graduates. She published over 130 articles in international conference proceedings and journals with four international book chapters. She has rich project management experiences as a project leader and a former technical committee chairs to the Thailand Government bodies in Telecommunications and ICT. She is very active in the international arena with the leadership positions in the international network, such as JICA Project for AUN/SEED-Net, and the professional organizations, such as the IEEE, IEICE, APSIPA, and ITU. She is currently a member of the IEEE Educational Activities Board (EAB) and the Chair of the IEEE EAB Pre-University Education Coordination Committee. She is also a member of the Board of Governors of the IEEE Consumer Electronics Society, from 2019 to 2021. She formerly led Educational Activities and Women in Engineering for the IEEE Asia Pacific (Region 10), from 2011 to 2016.

NICU SEBE (Senior Member, IEEE) is currently a Professor at the University of Trento, Italy, leading the research in the areas of multimedia information retrieval and human behavior understanding. He is a Fellow of the International Association for Pattern Recognition. He was the General Co-Chair of the IEEE FG Conference 2008 and ACM Multimedia 2013, and the Program Chair of the International Conference on Image and Video Retrieval, in 2007 and 2010, and ACM Multimedia 2007 and 2011. He was the Program Chair of ICCV 2017 and ECCV 2016, and the General Chair of ACM ICMR 2017.