ABSTRACT

Object detection generally shows promising results only using spatial information, but foreground object detection in visual surveillance requires proper use of temporal information in addition to spatial information. Recently, deep learning-based visual surveillance algorithms have shown improved results, in an environment similar to training one, compared to traditional background subtraction (BGS) algorithms. However, in unseen environments, they show poor performance compared to BGS algorithms. This paper proposes an algorithm that improves performance in unseen environments by integrating spatial and temporal information. We propose a spatio-temporal fusion network (STFN) that extracts temporal and spatial information from 3D and 2D networks. Also, we propose a method for stable training of the proposed STFN using a semi-foreground map. STFN can generate a compliant background model image and operate in real-time on a desktop with GPU. The proposed algorithm performs well in an environment different from training and is demonstrated by experiments using various public datasets.

INDEX TERMS
Visual surveillance, deep learning, foreground object detection, spatio-temporal information.

I. INTRODUCTION
Visual surveillance aims to detect foreground objects stably in various environments. Unlike general object detection, which usually considers a single image, foreground object detection in visual surveillance requires a proper assessment of both spatial and temporal information. Traditional BGS algorithms use background model images generated by statistical analysis and feedback. They usually detect foreground objects by comparing a current image with the background model image. Temporal information is reflected in the background model image by the update process. But, the reflection of spatial data is limited to the local area due to the fixed size of the window.

Recently, deep learning-based algorithms for visual surveillance show superior performance compared to BGS algorithms. FgSegNet_v2 [1] currently records the highest performance in the ranking of the CDnet2014 dataset [2]. It shows an average false detection rate of 0.2% in the entire image sequence by only using 1% of images for training. However, since FgSegNet_v2 [1] is a spatial network that processes only a single image without considering temporal information, it shows a performance drop in unseen environments.

This paper proposes a spatio-temporal fusion network (STFN) to perform better than BGS algorithms in unseen environments. We use the latest deep learning models, such as 3DCD [3] and FgSegNet_v2 [1], as a temporal and spatial network for the configuration of the proposed STFN. Also, we propose a method that uses a semi-foreground map for training, providing improved performance. In experiments, we follow the procedure proposed by Mandal and Vipparthi [4] to show performance improvement in unseen environments. They propose scene-independent data division (SIE)
and scene-dependent data division (SDE). The SIE setup uses different scenes for training and evaluation. The SDE setup divides the same scene into training and evaluation. We evaluate the generalization ability through the SIE setup in experiments.

The proposed algorithm has the following contributions.

1) It has excellent foreground object detection ability in a new environment not observed in training. It is possible by integrating a temporal network that extracts spatio-temporal information from multiple images and a spatial network that focuses on extracting foreground objects. The temporal network produces a background model by processing temporal information among successive images. The spatial network detects foreground objects by processing a difference image produced by subtracting a current image and a background model image.

2) We propose a method that uses loss derived from the semi-foreground map (SFM) for stable training and performance improvement of the STFN model.

II. RELATED WORKS
A. TRADITIONAL APPROACHES
Stauffer and Grimson [5] proposed a mixture of Gaussian (MOG) algorithm, which models the brightness change of pixels as the combination of multiple Gaussian distributions. Elgammal et al. [6] proposed an algorithm that uses kernel density estimation under the probabilistic non-parametric method. Kim et al. [7] proposed an algorithm that uses a codebook. They used intensity, color, and temporal features to make codewords, and later they were used to build a codebook. Barnich and Droogenbroeck [8] used a sample-based method for background modeling. They used previous frames samples to determine the foreground and background by comparing them to the current pixel.

St-Charles and Bilodeau [9] also proposed an improved algorithm of Barnich et al. [8] that used local binary similarity patterns [10] as additional features of pixel intensities and the modification of the update mechanism of thresholds and background model. Hanies and Xiang [11] proposed a new BGS algorithm based on Dirichlet process Gaussian mixture models that estimate background distributions per pixel. It is possible to detect a stable foreground per pixel using the non-parametric Bayesian method. St-Charles et al. [12] proposed a SuBSENSE algorithm that provides universal pixel-level segmentation using spatiotemporal binary features and color information. They introduced pixel-level feedback loops for the dynamic adjustment of internal parameters in an automated fashion. St-Charles et al. [13] proposed a visual surveillance algorithm based on non-parametric, pixel-level background modeling using word dictionaries drawn from traditional codebooks and sample consensus approaches.

Laugraud et al. [14] proposed LaBGen, a background modeling algorithm that includes a temporal median filter based on pixel units and a patch selection mechanism based on motion detection. Berjón et al. [15] proposed a non-parametric real-time and high-quality moving object detection strategy. They used both foreground and background models to cope with an environment in which the shapes of the background object and the foreground object are similar. Ortego et al. [16] proposed a post-processing framework that improves the foreground object detection performance of the existing BGS algorithms. Yang et al. [17] offered spatiotemporally scalable matrix recovery (SSMR) for separating foreground objects in environments with complex foreground objects and camera movements. Garg et al. [18] suggested a new block-based feature suitable for background modeling road environments and detecting moving vehicles. Hossain et al. [19] proposed Fast-D, a BGS-based foreground object detection algorithm. They used a non-smoothing color feature to make foreground object detection robust. Roy and Bouwmans [20] proposed a new pixel-based object detection algorithm using dual-type pixel-level information to generate a background model. Li et al. [21] proposed a generalized shrinkage thresholding operator (GSTO) that integrates three familiar shrinkage operators.

Traditional BGS algorithms have the advantage of not requiring training using ground-truth labels, but their performance is inferior to deep learning-based algorithms. This is because situations covered by statistical modeling are certainly smaller than that of deep learning-based algorithms.

B. DEEP LEARNING APPROACHES
Braham and Droogenbroeck [22] proposed a ConvNet that produces a foreground probability map with a LeNet-5 type network, which combines a background model image and a current image. This method performed better than other BGS methods in the CDnet2014 [2]. Zhao et al. [23] proposed a deep neural network consisting of two modules: background reconstruction and foreground segmentation. The background reconstruction part was trained in a supervised method. This method has weaknesses in new environments like Braham and Droogenbroeck [22].

Wang et al. [24] proposed multi-scale CNN (MSCNN) and MSCNN+Cascade models using multiple CNN models. For the cascade model, they used two CNN models. However, they did not prove generalization performance in a new environment. Zeng and Zhu [25] proposed a new multi-scale fully convolutional network (MFCN) based on VGG-16 [26]. They used RGB images as input without creating separate background model images. Babaei et al. [27] proposed a background pixel library (BL) that performs background modeling using the segmentation map of SuBSENSE [12]. They stacked the generated background model image and a current image depth-wise and used them as the CNN input. Lim and Keles [28] proposed the multi-scale segmentation architectures FgSegNet_M and FgSegNet_S. The FgSegNet_M down sampled an input image to two different scales. After that, an original image and the down sampled two images are passed through the CNN. FgSegNet_S extracted features of multiple scales by feature pooling.
module (FPM) with dilated convolutions. Lin et al. [29] proposed an algorithm that generates a foreground map using a deep neural network consisting of 20 convolutional layers and three deconvolutional layers. Zeng et al. [30] proposed an algorithm that uses the output of the semantic segmenter and BGS segmenter. They obtain an improved foreground map with post-processing and feed it back to the BGS algorithm. ICNET [31] and PSPNet [32] were used as semantic segmenters, and ViBe [8] and SuBSENSE [12] were used as BGS segmenters, showing superior performance compared to BGS algorithms under various conditions. However, it shows unsatisfactory performance compared to the latest deep-learning methods.

Qui and Li proposed [33] a fully convolutional encoder-decoder spatial-temporal network (FCESNet) that uses multiple images as inputs. FCESNet consists of an encoder, spatial-temporal information transmission (STIT) module, and decoder with multiple input/output structures. Gomaa et al. [34] proposed an algorithm that detects and tracks moving vehicles through background subtraction using CNN. Patil and Murala [35] proposed a motion saliency foreground network (MSFgNet). The model receives 50 images and generates a background model image through the background estimation module (GRBE). Lim et al. [1] proposed FgSegNet_v2 using a feature pooling module (FPM). They used the VGG-16 [26] as an encoder and compressed information as input of modified FPM (M-FPM).

Tezcan et al. [36] proposed a background subtraction algorithm for unseen videos called BSUV-Net. They show excellent performance in the SIE environment. But, an empty reference frame uses the intermediate value of initial frames. Therefore, it is impossible to immediately respond to the modified background environment where background objects are added or disappear. Akilan et al. [37] proposed a model composed of a 3D CNN-LSTM encoder and a 3D CNN-LSTM decoder. They proposed a double-encoding technique using an autoencoder-type micro-module. In addition, they used 3D convolutions to capture short temporal motions. Patil et al. [38] proposed an edge extraction mechanism (EEM) and dense residual block (DRB) for foreground object detection. Zheng et al. [39] proposed a new BGS algorithm based on parallel vision and Bayesian generative adversarial networks.

Rezaei et al. [40] proposed an algorithm for a generative low-dimensional background model (G-LBM) consisting of convolutional layers, fully connected layers, and deconvolutional layers. Mandal et al. [4] proposed ChangeDet, a lightweight and fast network suitable for visual surveillance. ChangeDet consists of depth reductionist background estimation (DRBE), contrasting feature assimilation (CFA), and contrasting feature-based encoder-decoder (CFE-CFD) blocks. This method shows superior generalization ability to other deep-learning methods in SIE and SDE environments.

Mandal et al. [3] proposed a fast and lightweight end-to-end 3D-CNN-based change detection network called 3DCD with excellent performance in a SIE environment. 3DCD consists of gradual reduction background estimation (GRBE), foreground saliency reinforcement (FSR), multi-schematic encoder-decoder (MScE-D), and compact foreground detection (CFD) blocks. Tezcan et al. [41] proposed a spatio-temporal data augmentation method for their previous work [36]. It included spatio-temporal crop, data amplification using illumination difference, and intermittent-object addition.

Li et al. [47] proposed an algorithm for object detection in video surveillance. They adjust the generic CNN-based classifier to detect objects in visual surveillance by transfer learning and module for effective learning of the local and global contexts in surveillance scenes. Nguyen et al. [48] deal with change detection in visual surveillance by integrating motion feature network and traditional BGS algorithms. The motion feature network uses features trained by a triplet network. Zhu et al. [49] proposed an algorithm to improve object detection in a nighttime scene. They proposed N2DGAN that generates a virtual daytime background model image corresponding to nighttime, improving object detection in nighttime images.

Deep learning-based visual surveillance algorithms [1], [35] have shown dramatic improvement compared to traditional BGS algorithms [11], [12], [15], [19]. But, they only perform well in scenes similar to training ones. In unseen environments, their performance is worse than conventional BGS algorithms. Some research [3], [4] has recently shown good performance in unseen environments, but still, there are significant margins for improvement.

III. PROPOSED METHOD

A. SPATIO-TEMPORAL FUSION NETWORK (STFN)

The proposed algorithm uses multiple current and past images as a network input to extract spatio-temporal information in image sequences. A temporal network is responsible for extracting information in the temporal domain, and it generates a background model image. A 2D network generates a foreground map by processing a difference image from the background model image and a current image. In this paper, the gradual reduction background estimation (GRBE) module in the 3DCD [3] and FgSegNet_v2 [1] are adopted as a temporal and spatial network in configuring the proposed STFN. The proposed algorithm effectively integrates these two networks and offers a method to train the proposed network.

Figure 1 shows the structure of the proposed STFN. We use the GRBE module to generate a background model image using 50 gray images. The generated background model image \((1 \times 224 \times 224 \times 1)\) is reshaped to a 3D tensor \((224 \times 224 \times 1)\) and then subtracted from a current image. Then, this subtracted image is used as input of FgSegNet_v2 to find foreground objects. The GRBE module plays the role of summarizing spatio-temporal data in successive images into a background model image. The original FgSegNet_v2 [1] uses a 3-channel RGB image as input and generates a 1-channel
foreground probability map. In the proposed model, we use 1-channel gray image as input. For this reason, we do not use the VGG-16 [26] pre-trained weights in FgSegNet_v2 [1]. Also, we suggest a loss term using a semi-foreground map (SFM) to improve the training process and performance. Figure 2(a) shows the GRBE module in 3DCD [3]. Figure 2(b) shows the structure of FgSegNet_v2. More details about the GRBE [3] and FgSegNet_v2 [1] can be found in the original papers. Although the quality of the background model image by the GRBE module is inferior to the SuBSENSE [12], we show that performance improvement is possible by integrating it with the FgSegNet_v2 within the end-to-end configuration. The proposed STFN model that uses the GRBE and FgSegNet_v2 as sub-modules overcomes their limitation.

B. LOSS WITH SEMI-FOREGROUND MAP
The proposed STFN contains two networks, which require equally trained well. If we train the proposed STFN model only using output, like most deep learning-based visual surveillance algorithms, a training imbalance between the GRBE and FgSegNet_v2 may occur. Even if we have a background model image with all pixel values of 0 as the output of the GRBE module, the FgSegNet_v2 module can be trained.
because the difference image is the same as the current image. We must train the GRBE module and FgSegNet_v2 module equally well for performance improvement. We solve this problem by adding a cost term that leverages output generated by the GRBE module.

The output of the GRBE module is a background model image. Therefore, a ground-truth background model image is required for computing loss. However, most visual surveillance datasets do not provide the ground-truth background model image since it is difficult to obtain. Fortunately, visual surveillance datasets provide ground-truth foreground maps. We propose a new method for loss computation leveraging the background model image estimated by the GRBE module and the ground-truth foreground map. Since the background model image and the foreground map belong to different domains, it is challenging if we compute loss directly by comparing them. For this reason, in this paper, we produce a semi-foreground map (SFM) by difference operation between the estimated background model image and the original image. We use SFMs to compute additional loss to train the GRBE module.

The additional loss computation for the GRBE module is based on the following assumptions. (1) A relatively significant brightness difference between a current image and a background model image exists at foreground objects. (2) A relatively slight brightness difference exists in background regions between a current image and a background model image. (3) We can use one background model image as a representative background model image in a short time interval. Under the above assumptions, we obtain SFMs by computing difference images between the estimated background model image and multiple input images. We compute loss using SFMs and ground-truth foreground maps. Since one background model image is generated using successive original images by the GRBE module, it is used for all input images when obtaining SFMs. Although the presented cost computation has limitations in dynamic backgrounds, static foreground objects, and illumination changes, we show that using SFMs help train the STFN model effectively through experimental results.

We configure the SFM generation process with no trainable parameters by layer difference and nonlinear functions without adding convolutional layers. If we have trainable parameters in the SFM generation, high-quality SFM generation is possible even if the GRBE module generates a poor background model image.

The appropriate nonlinear function at the output stage of the SFM is selected based on the following considerations. In general, a foreground map has a value between 0 and 1. Therefore, we can choose either \textit{tanh} or sigmoid as candidates, excluding ReLU, which produces an infinite output range. Since we take the absolute value on difference images, the nonlinear function receives a value larger than 0 as input. Under these conditions, the sigmoid has an output value between 0.5 and 1. Therefore, we choose \textit{tanh} as a suitable nonlinear function for SFM generation.

In \textit{tanh}, the derivative value decreases sharply as the value increases from 0. In the case of an 8-bit image, the brightness has a value between 0 and 255. A difference image after absolute operation also has a value between 0 and 255. If we directly use such an input range value, the gradient of \textit{tanh} has a value close to 0 for most input ranges which may cause problems during training. Therefore, it is necessary to adjust the input range to a region where a gradient value of \textit{tanh} is larger than zero. We use a pre-processing method that divides the pixel value of the input image by a value K larger than one, which guarantees the gradient value of \textit{tanh} has a non-zero value. Throughout this pre-processing step, we can cope with the problem that the gradient approaches zero during back-propagation.

We apply the following pre-processing for all pixels of the current and multiple past images used as inputs of the proposed STFN model.

\[
\tilde{I}(x, y, t) = \frac{I(x, y, t)}{K} \quad (1)
\]

\(I(x, y, t)\) represents the brightness value at location \((x, y)\) at time \(t\) on an image, and \(K\) is a real number. \(\tilde{I}(x, y, t)\) represents the brightness value at location \((x, y)\) at time \(t\) after pre-processing. Next, we take the difference per pixel between a background model image generated by the GRBE module and input images.

\[
I_d^{GRBE}(x, y, t - i) = I_b^{GRBE}(x, y, t) - \tilde{I}(x, y, t - i) \quad (2)
\]

\(I_d^{GRBE}\) and \(I_b^{GRBE}\) represents a background model image by the GRBE module and a difference image. The GRBE module estimates a background model image using previous \(N\) images. We obtain \(N\) difference images by Eq. (2). We convert difference images in Eq. (2) to foreground maps through the following process. Finally, we get \(N\) semi-foreground maps.

\[
I_f^{SFM}(x, y, t - i) = \tanh\left(\frac{I_d^{GRBE}(x, y, t - i)}{K}\right) \quad (3)
\]

\(I_f^{SFM}\) represents a semi-foreground map. A loss for training the proposed STFN consists of the following terms.

\[
L_{SFM}(i) = BCE\left(I_f^{SFM}(x, y, t - i), I_f^{GT}(x, y, t - i)\right) \quad (4)
\]

\(I_f^{GT}\) and \(I_f^{SFM}\) represents a ground-truth foreground and a foreground map by FgSegNet_v2. FFM represents the final foreground map. BCE\((a, b)\) is the binary cross-entropy of \(a\) and \(b\). The final loss is as follows.

\[
L = \alpha L_{SFM} + (1 - \alpha)L_{FFM} \quad (6)
\]

\(\alpha\) represents the weight of the two-loss items and has a value between 0 and 1. K and \(\alpha\) are hyperparameters, and we determine their value through experiments.
After training, the output of the FgSegNet_v2 module is used as the final output. The output of the GRBE module is only used in training the proposed STFN. Since incomplete assumptions are used for SFM generation, we cannot lower loss by SFM below a certain level. For this reason, it is challenging to check training progress by investigating the loss of SFM. Therefore, we investigate FFM loss to determine whether the training is going well.

IV. EXPERIMENTAL RESULTS

In visual surveillance, two methods are widely used for performance evaluation. The first method uses different scenes for training and evaluation. The second method divides images from the same sequence into training and evaluation. Mandal et al. [4] noted these two evaluation methods as a scene-independent data division (SIE) environment and a scene-dependent data division (SDE) environment. Visual surveillance systems deployed in various environments should be able to detect foreground objects stable without additional training. Therefore, this paper uses quantitative and qualitative evaluation in the SIE environment to investigate the generalization power.

In the proposed STFN model, we use 50 recent images selected among 196 frames with four frame intervals as inputs to the GRBE module. We configure SFM to have 50 images, like the number of input images. The FFM consists of one foreground probability map that matches the current image. We use Kingma and Adam [42] as an optimizer and set the learning rate to 0.001. If the validation loss does not decrease more than five times, we reduce the learning rate to half-time. We use Keras [43] for implementation.

Three datasets, such as the CDnet2014 [2], LASIESTA [44], and SBI [45], are used in experiments. Evaluation is done in two ways. First, evaluation is done on the entire SBI and LASIESTA datasets after training the model using the CDnet 2014 dataset. Second, evaluation is done after we make the SIE environment by separating the LASIESTA and SBI datasets.

We use the following as evaluation indicators: precision, recall, F-measure (FM), percentage of the wrong classification (PWC), false-positive rate (FPR), false-negative rate (FNR), and specificity (Sp).

\[
\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \\
\text{FM} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \\
\text{PWC} = \frac{TP + TN + FP + FN}{TP + FN} \times 100, \\
\text{FPR} = \frac{FP}{TN + FP}, \quad \text{FNR} = \frac{FN}{TP + FN}, \\
\text{Sp} = \frac{TN}{TN + FP}
\]

TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative.

### TABLE 1. FM score according to the value of K.

| Scene | K value | 1 | 10 | 25.5 | 51 | 127.5 | 255 |
|-------|---------|---|----|------|----|-------|-----|
| SI 2  | 0.2140  | 0.9672 | 0.9687 | 0.9603 | 0.9444 | 0.9288 |
| CA 2  | 0.2865  | 0.7507 | 0.6754 | 0.7004 | 0.7124 | 0.7462 |
| OC 2  | 0.2505  | 0.9409 | 0.9471 | 0.9330 | 0.9451 | 0.9250 |
| IL 2  | 0.2643  | 0.8908 | 0.9068 | 0.8934 | 0.8945 | 0.6602 |
| MB 2  | 0.6177  | 0.8484 | 0.8800 | 0.8417 | 0.8529 | 0.8595 |
| BS 2  | 0.1069  | 0.7110 | 0.7633 | 0.7256 | 0.7328 | 0.7285 |
| CL 2  | 0.0402  | 0.9477 | 0.9593 | 0.9537 | 0.9349 | 0.9455 |
| RA 2  | 0.3821  | 0.9761 | 0.9788 | 0.9719 | 0.9696 | 0.9625 |
| SN 2  | 0.0136  | 0.8338 | 0.7831 | 0.8367 | 0.6807 | 0.6975 |
| SU 2  | 0.0615  | 0.9290 | 0.9291 | 0.9322 | 0.9291 | 0.9191 |
| Mean(2 scenes) | 0.2237 | 0.8781 | 0.8791 | 0.8750 | 0.8584 | 0.8373 |
| Mean(1 scenes) | 0.9800 | 0.9969 | 0.9964 | 0.9964 | 0.9957 | 0.9331 |

### TABLE 2. FM score according to the value of weight $\alpha$.

| Scene | $\alpha$ value | 0 | 0.1 | 0.5 | 0.9 |
|-------|----------------|---|-----|-----|-----|
| SI 2  | 0.6829 | 0.9687 | 0.9681 | 0.9636 |
| CA 2  | 0.3617 | 0.6754 | 0.7312 | 0.7183 |
| OC 2  | 0.3838 | 0.9471 | 0.9114 | 0.8793 |
| IL 2  | 0.4482 | 0.9068 | 0.8982 | 0.8792 |
| MB 2  | 0.6747 | 0.8800 | 0.8523 | 0.8540 |
| BS 2  | 0.1935 | 0.7633 | 0.7354 | 0.7324 |
| CL 2  | 0.0684 | 0.9593 | 0.9568 | 0.9414 |
| RA 2  | 0.2387 | 0.9788 | 0.9770 | 0.9760 |
| SN 2  | 0.0134 | 0.7831 | 0.7819 | 0.7880 |
| SU 2  | 0.1015 | 0.9291 | 0.9324 | 0.9212 |
| Mean(2 scene) | 0.3167 | 0.8791 | 0.8745 | 0.8654 |
| Mean(1 scene) | 0.9880 | 0.9964 | 0.9962 | 0.9947 |

### A. DETERMINATION OF HYPERPARAMETER

Hyperparameter used in the proposed method is K in Eq. (1) and $\alpha$ in Eq. (6). We determine their values by investigating experimental results. In the LASIESTA dataset [43], we used scene 1 for training and scene 2 for evaluation out of 10 categories.

Table 1 shows the FM scores according to the variation of the K value used for pre-processing. It indicates that scenes used for training provide high FM result values regardless of the K value, but the scenes not used for training provide significantly low FM results depending on the K value. It indicates that the proposed pre-processing method is necessary for robust object detection in an environment different from the training environment.

Table 2 shows FM scores according to the SFM loss weight $\alpha$ value variation. When the $\alpha$ value is 0, it corresponds to training the proposed STFN model without using SFM. In this case, the FM score of scenes used for training is 0.9924, which shows excellent performance. But, for scenes not used for training, the FM score is 0.2921, showing a dramatic performance drop. On the other hand, when we train the proposed STFN model using the cost term by SFM, we obtain excellent performance even for scenes not used for training. This shows the necessity of cost terms by SFM.
for stable training of the proposed STFN model. Based on Tables 1 and 2, K and α were set to 127.5 and 0.1.

B. EXPERIMENTAL RESULTS USING THE CDNET2014 DATASET

After training the proposed STFN model using the CDnet2014 dataset, we evaluate the performance of the proposed method using the LAEIESTA and SBI datasets. Among the 11 categories in the CDnet2014 dataset, we use scenes from five categories for training. Table 3 shows scenes used for training in the CDnet2014 dataset. We used 42,345 images from 23 scenes for training. We divide them into 80% for train and 20% for validation per scene. Finally, we used 33,868 images for training and 8,477 images for validation.

1) EVALUATION OF THE LAEIESTA DATASET

We use 20 scenes in 10 categories for evaluation in the LAEIESTA dataset. Table 4 shows the quantitative evaluation results of the proposed algorithm. Table 5 shows the comparison results of FM scores to other algorithms. We evaluate 3DCD [3] and FgSegNet_v2 [1] after training them using the same dataset as the proposed algorithm. SuBSENSE [12] and PAWCS [13] are evaluated using BGSLibrary [46]. We use evaluation statistics specified in the papers for the rest of the algorithms. If we do not use SFMs, we can notice that the performance of the proposed STFN model significantly degrades. The proposed algorithm performs best when the STFN model is trained with SFM loss.

Compared to the 3DCD [3] and FgSegNet_v2 [1], the proposed algorithm improves by 13.1% and 104.5%, respectively. FgSegNet_v2 provides excellent results in the SDE environment. Still, we notice large performance drops in the SIE environment, like the LAEIESTA dataset. The result of the SBI dataset shows a relatively small performance improvement compared to the LAEIESTA dataset.

In Table 7, the proposed algorithm gives a low performance in some categories in the SBI dataset. This is caused by wrong background model images, as shown in Figure 4. We think that improving the quality of the background model image would partially solve this problem.

C. EXPERIMENTAL RESULTS USING LAEIESTA AND SBI DATASET

The previous experiment was done as follows. First, the model was trained using images from the CDnet2014 dataset, and then the trained model was evaluated on the LAEIESTA and SBI datasets. In this case, the LAEIESTA and SBI dataset are divided into training and evaluation data, considering an environment where training data is small.

1) EVALUATION BY DIVIDING LAEIESTA DATASET

We divide the LAEIESTA dataset into training and evaluation. The division of the LAEIESTA dataset was done following 3DCD [3]. We use scene one for training and scene two in the assessment among ten categories of the LAEIESTA dataset. We use 4,300 images in experiments, where we use 3,440 images for training and 860 for validation.

Table 8 shows the comparison result of FM scores to other algorithms. Results noted in the 3DCD [3] were used as results of FgSegNet_v2 [1]. The proposed algorithm provides an 11.3% improvement compared to 3DCD [3] and ChangeDet [4], which show the best results among comparison algorithms.

2) EVALUATION BY DIVIDING SBI DATASET

An evaluation was done by dividing the SBI dataset into training and evaluation datasets. We experimented the same way in the 3DCD [3] and ChangeDet [4]. We use the ‘Candela’, ‘CAVIAR2’, ‘CaVignal’, and ‘Highway2’ scenes of the SBI dataset for evaluation and the remaining scenes for training.
We used 3,455 images in experiments where we used 2,761 images for training and 694 for validation. Table 9 shows comparison results to other algorithms. We use the results of 3DCD [3] and ChangeDet [4], as noted in the paper. Although the performance degrades compared to the LASIESTA dataset because there is less training data, the proposed algorithm shows the best result compared to other algorithms. In addition, we can notice that the generalization ability becomes insufficient when we do not use the SFM loss in the proposed algorithm. The proposed algorithm provides an 18.3% improvement over 3DCD [3], showing the best performance among comparison algorithms.

### D. Comparison of Background Model Image
In the proposed STFN model, we obtain a background model image by processing multiple images along the temporal dimension from the present to the past. Classical BGS algorithms and modern deep learning-based 3DCD [3] use background model images. The SBI dataset provides a ground-truth background model image, while the LASIESTA dataset does not. We manually create a background model image using the ground-truth foreground map of the LASIESTA dataset. Unlike other methods that provide color background model images, the proposed method and 3DCD [3] generate gray background model images.

|TABLE 4. | Various quantitative results of the proposed algorithm with the lasiesta dataset. |
|---|---|
|Categories | FM | PWC | Precision | Recall | FPR | FNR | Sp |
|I SI | 0.9422 | 0.3380 | 0.9756 | 0.9112 | 0.0007 | 0.0888 | 0.9993 |
|I CA | 0.8387 | 2.2871 | 0.8447 | 0.8351 | 0.0122 | 0.1669 | 0.9879 |
|I OC | 0.9654 | 0.1704 | 0.9627 | 0.9683 | 0.0010 | 0.0318 | 0.9990 |
|I IL | 0.9548 | 0.3176 | 0.9362 | 0.9126 | 0.0014 | 0.0874 | 0.9986 |
|I MB | 0.9166 | 0.8304 | 0.9351 | 0.9049 | 0.0029 | 0.0951 | 0.9971 |
|I BS | 0.8555 | 0.2489 | 0.8888 | 0.8278 | 0.0011 | 0.1722 | 0.9989 |
|O CL | 0.9574 | 0.3077 | 0.9659 | 0.9149 | 0.0060 | 0.0429 | 0.9995 |
|O RA | 0.9419 | 0.0665 | 0.8684 | 0.9731 | 0.0005 | 0.0269 | 0.9995 |
|O SU | 0.9455 | 0.0683 | 0.9631 | 0.9287 | 0.0002 | 0.0713 | 0.9998 |
|Average | 0.9197 | 0.4599 | 0.9262 | 0.9161 | 0.0021 | 0.0839 | 0.9979 |

|TABLE 5. | FM score comparison with other algorithms with the lasiesta dataset. |
|---|---|
|LASIESTA | Proposed (SIE) | Proposed (SIE) | MSFeNet (SDE+SIE) | 3DCD [3] | Fast-D [19] | SubSENSE [12] | Berjón et al. [15] | Haines et al. [11] | FgSegNe [1] |
|DATASET | (SIE) (w/o SFM) | (SIE) (w/o SFM) | [35] (SIE) | [3] (SIE) | [19] (Traditional Algorithm) | [12] (Traditional Algorithm) | [15] (Traditional Algorithm) | [11] (Traditional Algorithm) | [1] (SIE) |
|I SI | 0.9422 | 0.2997 | 0.9264 | 0.9076 | 0.9287 | 0.9086 | 0.8806 | 0.8876 | 0.7568 |
|I CA | 0.8387 | 0.4044 | 0.9213 | 0.6721 | 0.8924 | 0.8702 | 0.8444 | 0.8938 | 0.7194 |
|I OC | 0.9654 | 0.2301 | 0.9163 | 0.9559 | 0.9194 | 0.9249 | 0.7807 | 0.9223 | 0.3563 |
|I IL | 0.9242 | 0.2417 | 0.8967 | 0.9365 | 0.5021 | 0.4685 | 0.6488 | 0.8491 | 0.4628 |
|I MB | 0.9166 | 0.5306 | 0.9143 | 0.8776 | 0.9480 | 0.9173 | 0.9374 | 0.8440 | 0.6312 |
|I BS | 0.8555 | 0.2432 | 0.7157 | 0.8370 | 0.6182 | 0.6480 | 0.6644 | 0.6809 | 0.3825 |
|O CL | 0.9574 | 0.1839 | 0.8806 | 0.9064 | 0.9368 | 0.9153 | 0.9727 | 0.8267 | 0.4343 |
|O RA | 0.9361 | 0.3077 | 0.8659 | 0.8378 | 0.9378 | 0.8756 | 0.8670 | 0.8908 | 0.4912 |
|O SN | 0.9149 | 0.0668 | 0.8952 | 0.8461 | 0.8789 | 0.7925 | 0.7787 | 0.1750 | 0.0777 |
|O SU | 0.9455 | 0.0368 | 0.7869 | 0.8975 | 0.8710 | 0.7919 | 0.7222 | 0.8568 | 0.1861 |
|Average | 0.9197 | 0.2365 | 0.8717 | 0.8668 | 0.8482 | 0.8113 | 0.8051 | 0.7826 | 0.4498 |

|TABLE 6. | Various quantitative results of the proposed algorithm using the SBI dataset. |
|---|---|
|Scenes | FM | PWC | Precision | Recall | FPR | FNR | Sp |
|Board | 0.9468 | 2.9927 | 0.9955 | 0.9027 | 0.0017 | 0.0973 | 0.9983 |
|CIVIAR1 | 0.9598 | 0.3003 | 0.9575 | 0.9622 | 0.0017 | 0.0378 | 0.9983 |
|CIVIAR2 | 0.8832 | 0.3960 | 0.8438 | 0.9264 | 0.0007 | 0.0736 | 0.9993 |
|CaVirtual | 0.5670 | 9.6145 | 0.5970 | 0.5398 | 0.0481 | 0.4602 | 0.9519 |
|Candela | 0.7660 | 1.6145 | 0.9104 | 0.6612 | 0.0027 | 0.3388 | 0.9973 |
|Hall & Monitor | 0.9565 | 0.2027 | 0.9875 | 0.9274 | 0.0003 | 0.0726 | 0.9997 |
|Highway1 | 0.9079 | 1.8378 | 0.9020 | 0.9139 | 0.0109 | 0.0861 | 0.9981 |
|Highway2 | 0.9513 | 0.2973 | 0.9369 | 0.9661 | 0.0020 | 0.0339 | 0.9980 |
|Human Body2 | 0.9511 | 0.9735 | 0.9740 | 0.9292 | 0.0028 | 0.0708 | 0.9972 |
|IBM TEST2 | 0.9576 | 0.3967 | 0.9417 | 0.9740 | 0.0029 | 0.0260 | 0.9971 |
|People & Foliage | 0.6462 | 27.5422 | 0.9857 | 0.4807 | 0.0077 | 0.5193 | 0.9923 |
|Average | 0.8630 | 4.1698 | 0.9120 | 0.8349 | 0.0074 | 0.1651 | 0.9926 |
|Average (Except P&F scene) | 0.8847 | 1.8326 | 0.9046 | 0.8703 | 0.0074 | 0.1297 | 0.9926 |
images. Therefore, we present only qualitative evaluation results.

Figure 5 shows background model images by the proposed and other algorithms. Red rectangles represent incorrect areas in the background model image. BGS algorithms show unstable background model images at the initialization stage. In particular, it takes a long time to generate correct background model images when it contains foreground objects at the initial step.

The proposed method can cope with these problems and creates a background model image that complies with the general situation. However, the ghost objects phenomenon in BGS algorithms at the bootstrap environment also occurs in the proposed algorithm, as shown in the ‘CaVignal’ scene in Figure 5. The proposed algorithm gives a less accurate background model image when foreground objects are stopped for a long time, as shown in the ‘I_CA_2’ scene in Figure 5. Also, the proposed algorithm is limited in that it generates a grayscale background model image, while traditional BGS algorithms provide a three-channel RGB background model image. Although the background model image generated by the proposed algorithm has margins for improvements, we obtain improved performance by integrating it with the 2D spatial network.
TABLE 7. FM score comparison with other algorithms using the SBI dataset.

| SBI DATASET     | Proposed (SIE) | Proposed (SIE) (w/o SFM) | Yang et al. [17] (Traditional Algorithm) | 3DCD [3] (SIE) | PAWCS [13] (Traditional Algorithm) | SuBSENSE [12] (Traditional Algorithm) | FgSegNet_v2 [1] (SIE) |
|-----------------|----------------|--------------------------|------------------------------------------|----------------|-----------------------------------|---------------------------------------|------------------------|
| Board           | 0.9468         | 0.8814                   | 0.91                                      | 0.9187         | 0.7798                             | 0.5777                               | 0.6942                 |
| CAVIAR1         | 0.9598         | 0.6650                   | 0.95                                      | 0.9401         | 0.8589                             | 0.9144                               | 0.7920                 |
| CAVIAR2         | 0.8832         | 0.9974                   | 0.84                                      | 0.8735         | 0.6772                             | 0.8714                               | 0.0600                 |
| Candela         | 0.5670         | 0.7651                   | 0.83                                      | 0.5275         | 0.3697                             | 0.3980                               | 0.9191                 |
| Hall & Monitor  | 0.9565         | 0.4258                   | 0.83                                      | 0.8993         | 0.7411                             | 0.7758                               | 0.7592                 |
| Highway1        | 0.9079         | 0.6747                   | 0.72                                      | 0.7712         | 0.7015                             | 0.5523                               | 0.7759                 |
| Highway2        | 0.9513         | 0.6596                   | 0.95                                      | 0.9310         | 0.9031                             | 0.8937                               | 0.9174                 |
| Human Body2     | 0.9511         | 0.4183                   | 0.88                                      | 0.9157         | 0.7013                             | 0.8346                               | 0.6068                 |
| IBM TEST2       | 0.9576         | 0.6047                   | 0.89                                      | 0.8967         | 0.9386                             | 0.9390                               | 0.5357                 |
| People & Foliage| 0.6462         | 0.3805                   | -                                        | 0.7746         | 0.3162                             | 0.2660                               | 0.6845                 |
| Average         | 0.8630         | 0.5485                   | -                                        | 0.8110         | 0.7145                             | 0.6871                               | 0.6435                 |
| Average (Except | 0.8847         | 0.5653                   | 0.87                                      | 0.8146         | 0.7544                             | 0.7293                               | 0.6394                 |
| P&F scene)      |                |                          |                                          |                |                                   |                                      |                        |

FIGURE 4. The qualitative evaluation with the SBI dataset.

E. COMPARISON OF COMPUTATION TIME

In this study, an experiment was done using a PC with AMD R9 3900X CPU and NVIDIA RTX 2080Ti 11GB GPU. Table 10 compares the number of parameters and fps by the proposed algorithm, FgSegNet_v2 [1] and 3DCD [3].
TABLE 8. FM score comparison with other algorithms using the lasiesta dataset.

| LASIESTA DATASET | Proposed (with SFM) | Proposed (w/o SFM) | 3DCD [3] | ChangeDet [4] | Berjón et al. [15] | Haines et al. [11] | FgSegNet-v2 [11] (SIE) |
|------------------|---------------------|-------------------|---------|---------------|-------------------|-------------------|-------------------------|
| I_ST 2           | 0.9687              | 0.6829            | 0.86    | 0.83          | 0.8403            | 0.8130            | 0.53                    |
| I_CA 2           | 0.6754              | 0.3617            | 0.49    | 0.66          | 0.7826            | 0.8656            | 0.58                    |
| I_OC 2           | 0.9471              | 0.3838            | 0.93    | 0.89          | 0.8600            | 0.9526            | 0.25                    |
| I_IL 2           | 0.9068              | 0.4482            | 0.85    | 0.87          | 0.6523            | 0.8122            | 0.41                    |
| I_MB 2           | 0.8800              | 0.6747            | 0.79    | 0.77          | 0.9204            | 0.7064            | 0.63                    |
| I_BS 2           | 0.7633              | 0.1935            | 0.87    | 0.82          | 0.6156            | 0.7333            | 0.25                    |
| O_CL 2           | 0.9593              | 0.0684            | 0.87    | 0.90          | 0.9045            | 0.9588            | 0.54                    |
| O_RA 2           | 0.9788              | 0.2387            | 0.87    | 0.85          | 0.8886            | 0.9590            | 0.54                    |
| O_SN 2           | 0.7831              | 0.0134            | 0.49    | 0.51          | 0.6256            | 0.0426            | 0.05                    |
| O_SU 2           | 0.9291              | 0.1015            | 0.83    | 0.81          | 0.7669            | 0.9021            | 0.29                    |
| Average          | 0.8791              | 0.3167            | 0.79    | 0.79          | 0.7857            | 0.7746            | 0.41                    |

TABLE 9. FM score comparison with other algorithms using the SBI dataset.

| Scenes          | Proposed (w/o SFM) | Proposed (w/ SFM) | FgSegNet-S [28] | FgSegNet-M [28] | FgSegNet-v2 [1] | ChangeDet [4] | 3DCD [3] |
|-----------------|--------------------|-------------------|-----------------|-----------------|-----------------|----------------|---------|
| Candela         | 0.13               | 0.64              | 0.23            | 0.15            | 0.10            | 0.27           | 0.67    |
| CAVIAR2         | 0.07               | **0.86**          | 0.11            | 0.14            | 0.56            | 0.56           | 0.62    |
| CAVignal        | 0.62               | 0.51              | 0.68            | **0.72**        | 0.63            | 0.48           | 0.53    |
| Highway2        | 0.13               | **0.82**          | 0.24            | 0.21            | 0.58            | 0.64           | 0.59    |
| Avg             | 0.24               | 0.71              | 0.31            | 0.30            | 0.40            | 0.57           | 0.60    |

FIGURE 5. Comparison of the generated background model image with other algorithms.

The proposed algorithm shows higher generalization performance and requires a lower computational cost than 3DCD [3]. The proposed algorithm can process 20% faster than 3DCD [3]. In FgSegNet-v2 [1], computation is three times
TABLE 10. Comparison of computation time.

| Method          | Number of parameters | FPS       |
|-----------------|----------------------|-----------|
| FgSegNet v2 [1] | 9.23M (trainable: 7.49M) | 111       |
| 3DCD [3]        | 0.13M                | 30        |
| Proposed        | 9.24M                | 37        |

faster than the proposed algorithm. However, in the SIE environment, the generalization performance is lower than the proposed algorithm. The proposed algorithm can operate at 30fps, enough for real-time processing in visual surveillance.

V. CONCLUSION

In this paper, we have proposed a spatio-temporal fusion network (STFN) that integrates a 3D temporal network and a 2D spatial network to enhance object detection in visual surveillance. The temporal network provides a background model image by summarizing spatio-temporal information among consecutive images. Finally, the spatial network detects foreground objects by processing a difference image from a current image and a background model image. Also, we have proposed additional cost terms derived from a semi-foreground map for training the proposed STFN. We have shown that the proposed method can generate a stable background model image through other loss computations. The proposed STFN model performs better than the 3DCD [3] and FgSegNet_v2 [1], which we adopted for the configuration of the STFN. The proposed algorithm heavily depends on the quality of the background model image. Currently, the generated background model image is gray. If we could have a color background model image, it would have more potential to detect foreground objects. Also, the current algorithm has weaknesses when foreground objects stop long. In future studies, we will try to use RNN as a temporal network to cope with these problems.

REFERENCES

[1] L. A. Lim and H. A. Keles, “Learning multi-scale features for foreground segmentation,” Pattern Anal. Appl., vol. 23, no. 3, pp. 1369–1380, 2020.
[2] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, “CDNet 2014: An expanded change detection benchmark dataset,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, Jun. 2014, pp. 387–394.
[3] M. Mandal, V. Dhar, A. Mishra, S. K. Vipparthi, and M. Abdel-Mottaleb, “3DCD: Scene independent end-to-end spatiotemporal feature learning framework for change detection in unseen videos,” IEEE Trans. Image Process., vol. 30, pp. 546–558, 2021.
[4] M. Mandal and S. K. Vipparthi, “Scene dependency matters: An empirical study of scene dependent and scene independent evaluation for CNN-based change detection,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 3, pp. 2031–2044, Mar. 2020.
[5] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2, Jun. 1999, pp. 246–252.
[6] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” Proc. IEEE, vol. 90, no. 7, pp. 1151–1163, Jul. 2002.
[7] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, “Real-time foreground–background segmentation using codebook model,” Real-Time Imag., vol. 11, no. 3, pp. 172–185, Jun. 2005.
[8] O. Barnich and M. Van Droogenbroeck, “ViBe: A universal background subtraction algorithm for video sequences,” IEEE Trans. Image Process., vol. 20, no. 6, pp. 1709–1724, Jun. 2011.
[9] P.-L. St-Charles and G.-A. Bilodeau, “Improving background subtraction using local binary similarity patterns,” in Proc. IEEE Winter Conf. Appl. Comput. Vis., Mar. 2011.
[10] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier, “Change detection in feature space using local binary similarity patterns,” in Proc. Int. Conf. Comput. Robot Vis., May 2013, pp. 106–112.
[11] T. S. F. Haines and T. Xiang, “Background subtraction with Dirichlet process mixture models,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 4, pp. 670–683, Apr. 2014.
[12] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, “SuBSENSE: A universal change detection method with local adaptive sensitivity,” IEEE Trans. Image Process., vol. 24, no. 1, pp. 359–373, Jan. 2015.
[13] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, “Universal background subtraction using word consensus models,” IEEE Trans. Image Process., vol. 25, no. 10, pp. 4768–4781, Oct. 2016.
[14] B. Laugrund, S. Piérard, and M. Van Droogenbroeck, “LaBG: A method based on motion detection for generating the background of a scene,” Pattern Recognit. Lett., vol. 96, pp. 12–21, Sep. 2017.
[15] D. Berjón, C. Cuevas, F. Morán, and N. García, “Real-time nonparametric background subtraction with tracking-based foreground update,” Pattern Recognit., vol. 74, pp. 156–170, Feb. 2018.
[16] D. Ortego, J. C. Sammiguel, and J. M. Martinez, “Hierarchical improvement of foreground segmentation masks in background subtraction,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 6, pp. 1645–1658, Jun. 2019.
[17] J. Yang, W. Shi, H. Yue, K. Li, J. Ma, and C. Hou, “Spatiotemporally scalable matrix recovery for background modeling and moving object detection,” Signal Process., vol. 168, Mar. 2020, Art. no. 107362.
[18] K. Garg, N. Ramakrishnan, A. Prakash, and T. Srikantan, “Rapid and robust background modeling technique for low-cost road traffic surveillance systems,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 5, pp. 2204–2215, May 2020.
[19] M. A. Hossain, M. I. Hossain, M. D. Hossain, N. T. Tho, and E.-N. Huh, “Fast-D: When non-smoothing color feature meets moving object detection in real-time,” IEEE Access, vol. 8, pp. 186756–186772, 2020.
[20] S. M. Roy and T. Bouwmans, “Dual information-based background model for moving object detection,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Oct. 2020, pp. 3219–3223.
[21] L. Li, Z. Wang, Q. Hu, and Y. Dong, “Adaptive nonconvex sparsity based background subtraction for intelligent video surveillance,” IEEE Trans. Ind. Informat., vol. 17, no. 6, pp. 4168–4178, Jun. 2021, doi: 10.1109/TII.2020.3009111.
[22] M. Brahmi and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks,” in Proc. IEEE Int. Conf. Syst. Signals Image Process., Jul. 2016, pp. 1–4.
[23] X. Zhao, Y. Chen, M. Tang, and J. Wang, “Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network,” in Proc. IEEE Int. Conf. Multimedia Expo. (ICME), Jul. 2017, pp. 343–348.
[24] Y. Wang, Z. Luo, and P.-M. Jodoin, “Interactive deep learning method for segmenting moving objects,” Pattern Recognit. Lett., vol. 96, pp. 66–75, Sep. 2017.
[25] D. Zeng and M. Zhu, “Background subtraction using multiscale fully convolutional network,” IEEE Access, vol. 6, pp. 16010–16021, 2018.
[26] K. Simonony and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.
[27] M. Babae, T. T. Dinh, and G. Rigoll, “A deep convolutional neural network for video sequence background subtraction,” Pattern Recognit., vol. 76, pp. 635–649, Apr. 2018.
[28] L. A. Lim and H. Yalim Keles, “Foreground segmentation using convolutional neural networks for multiscale feature encoding,” Pattern Recognit. Lett., vol. 112, pp. 256–262, Sep. 2018.
[29] C. Lin, B. Yan, and W. Tan, “Foreground detection in surveillance video with fully convolutional semantic network,” in Proc. 25th IEEE Int. Conf. Image Process. (ICIP), Oct. 2018, pp. 4118–4122.
[30] Y. Wang, L. Yuan, M. Zhu, M. Goeiale, and A. Kuijper, “Background subtraction with real-time semantic segmentation,” IEEE Access, vol. 7, pp. 153869–153884, 2019.
[31] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, “ICNet for real-time semantic segmentation on high-resolution images,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 405–420.
J.-Y. Kim, J.-E. Ha: Foreground Object Detection in Visual Surveillance With Spatio-Temporal Fusion Network

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2881–2890.

[33] M. Qiu and X. Li, “A fully convolutional encoder–decoder spatial–temporal network for real-time background subtraction,” IEEE Access, vol. 7, pp. 85949–85958, 2019.

[34] A. Gomaa, M. M. Abdelwahab, M. Abo-Zahhad, T. Minematsu, and R.-I. Taniguchi, “Robust vehicle detection and counting algorithm employing a convolution neural network and optical flow,” Sensors, vol. 19, no. 20, p. 4588, Oct. 2019.

[35] P. W. Patil and S. Murala, “MSFgNet: A novel compact end-to-end deep network for moving object detection,” IEEE Trans. Intell. Transp. Syst., vol. 20, no. 11, pp. 4066–4077, Nov. 2019.

[36] O. Tezcan, P. Ishwar, and J. Konrad, “BSUV-Net: A fully-convolutional neural network for background subtraction of unseen videos,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., Mar., Mar. 2020, pp. 2774–2783.

[37] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, “A 3D CNN-LSTM-based image-to-image foreground segmentation,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 3, pp. 959–971, Mar. 2020.

[38] P. W. Patil, K. M. Biradar, A. Dudhane, and S. Murala, “An end-to-end edge aggregation network for moving object segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 8149–8158.

[39] W. Zheng, K. Wang, and F.-Y. Wang, “A novel background subtraction algorithm based on parallel vision and Bayesian GANs,” Neurocomputing, vol. 394, pp. 178–200, Jun. 2020.

[40] B. Rezaei, A. Farnoosh, and S. Ostadabbas, “G-LBM: Generative low-dimensional background model estimation from video sequences,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Cham, Switzerland: Springer, Aug. 2020.

[41] M. O. Tezcan, P. Ishwar, and J. Konrad, “BSUV-Net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction,” 2021, arXiv:2101.09585.

[42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[43] F. Chollet. (2015). Keras. [Online]. Available: https://github.com/fchollet/keras

[44] C. Cuevas, E. M. Yez, and N. García, “Labeled dataset for integral evaluation of moving object detection algorithms: LASIESTA,” Comput. Vis. Image Understand., vol. 152, pp. 103–117, Nov. 2016.

[45] L. Maddalena and A. Petrosino, “Towards benchmarking scene background initialization,” in Proc. Int. Conf. Image Anal. Process., Genoa, Italy: Springer, 2015, pp. 469–476.

[46] A. Sobral, “BGSLibrary: An openCV C++ background subtraction library,” in Proc. 9th Workshop Visao Computacional, vol. 27, 2013, p. 24.

[47] X. Li, M. Ye, Y. Liu, and C. Zhu, “Adaptive deep convolutional neural networks for scene-specific object detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 9, pp. 2538–2551, Sep. 2019.

[48] T. P. Nguyen, C. C. Pham, S. V.-U. Ha, and J. W. Jeon, “Change detection by training a triplet network for motion feature extraction,” IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 2, pp. 433–446, Feb. 2019.

[49] Z. Zhu, Y. Meng, D. Kong, X. Zhang, Y. Guo, and Y. Zhao, “To see in the dark: N2DGAN for background modeling in nighttime scene,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 2, pp. 492–502, Feb. 2021.

JAE-YEUL KIM (Graduate Student Member, IEEE) received the B.S. and M.E. degrees in mechanical and automotive engineering from the Graduate School of Automotive Engineering, Seoul National University of Science and Technology, Seoul, South Korea, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree in electrical engineering and computer science with the Graduate School of Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology (DGIST). His current research interests include visual surveillance using deep learning and scene understanding for autonomous navigation.

JONG-EUN HA received the B.S. and M.E. degrees in mechanical engineering from Seoul National University, Seoul, South Korea, in 1992 and 1994, respectively, and the Ph.D. degree in mechanical engineering from KAIST, Daejeon, South Korea, in 2000. From February 2000 to August 2002, he worked at Samsung Corning, developing an algorithm for a machine vision system. From 2002 to 2005, he worked in multimedia engineering at Tongmyong University. Since 2005, he has been a Professor with the Department of Mechanical and Automotive Engineering, Seoul National University of Science and Technology. His current research interests include deep learning, intelligent robots, and vehicles.

* * *