Research Article

Foreground Detection Based on Superpixel and Semantic Segmentation

Junying Feng,1,2 Peng Liu,1 and Yong Kwan Kim2

1School of Intelligent Manufacturing, Weifang University of Science and Technology, Shandong, Weifang 261000, China
2Department of Information and Communication Engineering, Hoseo University, Chungcheongnam-do, Asan 31499, Republic of Korea

Correspondence should be addressed to Yong Kwan Kim; ykkim@hoseo.edu

Received 31 May 2022; Accepted 5 July 2022; Published 31 August 2022

Academic Editor: Jun Ye

Copyright © 2022 Junying Feng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Foreground detection is an essential step in computer vision and video processing. Accurate foreground object extraction is crucial for subsequent high-level tasks such as target recognition and tracking. Although many foreground detection algorithms have been proposed, foreground detection in complex scenes is still a challenging problem. This paper presents a foreground detection algorithm based on superpixel and semantic segmentation. It first uses multiscale superpixel segmentation to obtain the initial foreground mask. At the same time, a semantic segmentation network is applied to separate potential foreground objects, and then use the defined rules to combine the results of superpixel and semantic segmentation to get the final foreground object. Finally, the background model is updated with the refined foreground result. Experiments on the CDNet2014 dataset demonstrate the effectiveness of the proposed algorithm, which can accurately segment foreground objects in complex scenes.

1. Introduction

Foreground detection is a fundamental lower-level task of computer vision. The performance of foreground detection directly affects the performance of a series of subsequent assignments, like object recognition and tracking. The primary purpose of foreground detection is to extract foreground objects from video frames and obtain feature information of foreground objects. Foregrounds are usually moving objects, such as moving human bodies and vehicles. The foreground detection process can be regarded as a classification process of image pixels, looking for differences in continuous image sequences (i.e., videos) and extracting moving objects of interest according to the differences. The moving objects of interest are in the foreground, and the rest are classified as background. The foreground object detection is generally realized by background subtraction, which usually contains unsupervised and supervised algorithms. In recent surveys [1–4], we can find various background subtraction algorithms studied from multiple aspects. Despite the continuous development of computer vision-related technologies, foreground detection has achieved impressive results in some application scenarios. However, foreground detection still faces significant challenges in real-world applications due to various complex interferences in natural scenes, such as dynamic background, shadow, lousy weather, camouflage. From Figure 1, we note that traditional methods often make false positives and negatives.

Superpixel is an image segmentation technology proposed and developed by Ren and Malik [5], which refer to irregular pixel blocks with specific visual significance composed of adjacent pixels with similar texture, color, brightness, and other features. The pixels in a superpixel have a more robust connection with each other than pixels in a region. Some applications [6, 7] based on superpixels verify its effectiveness in foreground detection. Semantic segmentation is a deep learning algorithm that associates a label or class with each pixel of an image. It identifies the collection of pixels that make up a different class. It is exemplified in Figure 2. Semantic segmentation shows powerful performance in foreground detection [8]. Based on these
motivations, a new approach named superpixel and semantic segmentation foreground detection (SSSFD) is proposed to address the challenges mentioned above. SSSFD can accurately detect moving objects according to the experiments on the CDNet2014 dataset [9]. The contributions of this paper are the following: (1) A background subtraction algorithm for multiscale superpixel segmentation with postoptimization is proposed. (2) A semantic segmentation network is trained on the CDNet2014 subset. (3) The final results combining superpixel and semantic segmentation participate in the background model update in real-time.

The rest of the paper is organized as follows: Section 2 introduces the related works of our proposed method. In Section 3, we report the implementations of SSSFD in detail. The experiments and experimental results’ qualitative and quantitative analysis are described in Section 4, and finally, this paper is briefly summarized in Section 5.

2. Related Works

Over the past 30 years, plenty of foreground detection approaches have been studied. They can be grouped into unsupervised and supervised methods.

2.1. Unsupervised Methods. Unsupervised background subtraction algorithm usually combines handcraft features with advanced background modeling rules to achieve foreground separation. The unsupervised background subtraction algorithm usually consists of three parts: the initialization of the background model, the extraction of the foreground region, and the update of the background model.

The first unsupervised algorithm widely used for foreground detection was the Gaussian Mixture Model (GMM) proposed by Stauffer and Grimson [10], which is a statistical-based parametric model. The basic principle of GMM is to represent the distribution of image pixels in the time domain with multiple (3–5) Gaussian models. The central assumption of the GMM is that the pixel obeys a Gaussian distribution. The variance of a pixel determines the pixel class. If the variance of the pixel is higher than a specific value, the pixel is considered a foreground pixel. This method assumes that the variance of the background pixels is in a relatively low position, which is not always accurate in practice. In addition, when the background changes rapidly, it is hard for this method to keep up with the background variations, and the detection of fast-moving objects will result in missed detection and ghost. Moreover, efficiently updating model parameters to match changes in the background is also a challenging problem.

To overcome the problems arising from the above GMM, nonparametric statistical background models were proposed to improve foreground detection performance effectively. The nonparametric estimation background subtraction methods do not need to preassume the model and estimate
the model's parameters. The classic methods are Kernel Density Estimation (KDE) [11] and Visual Background Extractor (ViBe) [12]. KDE reads the historical data in the recent samples of each pixel. It calculates the actual probability density of a pixel from the kernel density estimation of the pixel for foreground detection. This method requires a large amount of calculation, the real-time performance is not good enough, and the detection accuracy is average. In some cases, a smear phenomenon will occur. The ViBe is a pixel-level foreground detection algorithm. The key idea of this method is to store a sample set for each pixel in the image. The sample set consists of the past pixel value of the pixel position and the pixel value of its neighborhood. Determine whether the pixel is a foreground or background point by comparing the sample set of the current pixel. If the pixel meets the characteristics of the sample set, the pixel is the background point. Otherwise, it is the foreground point. The ViBe model uses a random strategy to establish and maintain the background model. ViBe makes use of a time sampling update strategy to reduce the update frequency. This method requires less computation, less memory usage, and faster processing speed but has the disadvantage of ghost. In order to eliminate ghosts and further improve the performance of the ViBe, various algorithms based on ViBe have been proposed. St-Charles et al. [13] presented a pixel-level segmentation method with spatiotemporal binary features and color information. In addition, an adaptive feedback mechanism was used to dynamically adjust parameters such as model fidelity, segmentation noise level, distance threshold, and local update rate. This approach achieved effective and robust results in the most challenging scenes. Lu and Lu [14] proposed a weight-sample-based method using samples with variable weights, minimum-weight update policy, reward-and-penalty weighting strategy, and spatial-diffusion policy for foreground detection. Nonparametric models do not need to specify the distribution model of pixels but still need to use a threshold-based method for each pixel. Due to the different characteristics of various scenes, it is difficult to infer the threshold of each pixel through a unified calculation model.

2.2. Supervised Methods. Supervised background subtraction algorithms typically rely on a trained deep neural network (DNN) to segment the foreground from the background. All of these methods are divided into four steps: (1) use a particular algorithm to get a clean background image; (2) generate a dataset of specific scenes for training; (3) exploit the dataset to train the deep neural network; (4) utilize the trained neural network and the background image to perform foreground detection on the image sequence. Supervised methods can achieve good results for foreground detection tasks in many complex scenes, significantly improving the accuracy of the foreground detection.

Braham and Van Droogenbroeck [15] proposed the first background subtraction model ConvNet based on convolutional neural networks (CNNs). Temporal median filtering generates a static background image from 150 consecutive frames. A pair of 27×27 image patches is extracted from the background image and the corresponding original image and are input into the constructed CNN model. The output is the foreground probability of the center pixel in this input image patch. Because ConvNet is one of the most straightforward approaches to model the differences between the background and the foreground using CNNs, there are several limitations listed below: (1) the high-level information is challenging to learn through patches [16]; (2) the trained network only works in specific scenes due to the highly redundant data of video frame sequences in training. If the ConvNet is used for other video scenarios, it should be retrained; (3) since ConvNet adopts a pixelwise scheme, treating each pixel as independent, isolated false negatives and false positives may appear in the foreground mask; (4) ConvNet is a deep encoder-decoder network with a generator network. However, one of the drawbacks of this generator network is that the object edges cannot be preserved, leading to the generation of blurry foreground areas. Since the first valuable research, various subsequent algorithms have been proposed to alleviate these limitations. Wang et al. [17] designed a new background model based on CNN with two critical ideas. One is the multiscale structure. For the model to segment foreground objects of different sizes accurately, the original image is down-sampled with 0.75 and 0.5, respectively. The sampled image and the original image are used as the model's input. The second is a cascaded architecture. In order to reduce pixel misclassification caused by local information, the foreground probability map output by the first multiscale structure and the original image is re-input to the second multiscale structure. The second CNN model further calculates the accurate foreground probability map.

A fully convolutional network (FCN) was first proposed to process the semantic segmentation of an image by Long et al. [18]. The FCN consists entirely of convolutional layers, which convert the fully connected layers of CNNs into deconvolutional layers, thereby preserving the location and semantic information of pixels. FCN can efficiently learn to make dense predictions for per-pixel tasks. Therefore, it is widely used in pixel-by-pixel computer vision tasks such as pixel classification. Based on the traditional FCN with atrous convolution, pyramid scene parsing network (PSPNet) which was added to the PSP module between the encoder and the decoder was presented by Zhao et al. [19]. The PSP module can obtain more global context information, producing high-quality results in scene parsing assignments. DeepLab v1 [20] combines deep convolutional neural networks with conditional random field (CRF), which overcomes the local properties of deep networks and can better locate segmentation boundaries. DeepLab v2 [21] makes full use of atrous convolution, which can effectively expand the receptive field and incorporate more context without increasing parameters. In addition, deep convolution neural networks (DCNNs) are combined with CRF to optimize the network performance further. Moreover, the atrous spatial pyramid pooling (ASPP) module enhances the robustness of the network for multiclass segmentation at multiple scales. It uses different sampling scales and receptive fields to extract input features and capture target and context information at
multiple scales. DeepLab v3 [22] designs cascaded or parallel atrous convolution modules and improves ASPP. DeepLab v3+ [23] optimizes segmentation results by adding a simple yet efficient decoder module on top of DeepLab v3. Badrinarayanan et al. [24] proposed a deep network SegNet for semantic segmentation based on FCN and modified VGG-16 [25]. It still adopts the encoder–decoder structure, and the encoder part uses the first 13 layers of a convolutional network of VGG-16. Each encoder layer corresponds to one decoder layer. The output of the final decoder is fed into a softmax classifier, which generates class probabilities for each pixel independently.

In addition to the algorithms mentioned above, other supervised algorithms such as generative adversarial networks (GANs) [26–28], deep CNNs [29–32], structured CNNs [33], and 3D CNNs [34–36] have also achieved good results in foreground detection applications.

3. Proposed Approach: SSSFD

SSSFD combines an unsupervised algorithm based on multiscale superpixel segmentation (MSS) and a supervised algorithm based on semantic segmentation. The detailed implementation steps are introduced in this section. Figure 3 shows the pipeline of SSSFD. First, train the semantic segmentation network and then, the background subtraction MSS is performed on the input image to obtain the coarse foreground mask. The process mainly consists of multiscale superpixel segmentation, foreground detection, and post-processing. At the same time, the semantic foreground segmentation map is obtained by semantically segmenting the input image using the pretrained network. Subsequently, the background subtraction and semantic segmentation results are combined with specific rules to get refined foreground result. Finally, the background model is updated with the elegant foreground result.

3.1. Unsupervised Method MSS

3.1.1. Preprocessing. We first apply a Gaussian filter to the current input image. Gaussian filtering is a process of weighted averaging of the entire image. Each Gaussian filtered image pixel value is obtained by the weighted average of the original image pixel and its neighboring pixels. The template defines the weight and neighborhood.

Next, we use different scales to perform superpixel segmentation on the Gaussian filter image to get foreground probability maps at each scale. The number of pixels in a superpixel block, that is, how many superpixels an image is divided into, is significant. Suppose the size of the superpixel block is large. In that case, the pixels in the superpixel block may contain both foreground and background object pixels, which will reduce the accuracy and sensitivity of the final detection result. If the size of the superpixel block is small, compared with the pixelwise foreground detection algorithms, the accuracy of the last detection result is not significantly improved, and the robustness is poor. Therefore, we perform multiscale superpixel segmentation on video sequences and then execute foreground detection. We use the SLIC algorithm [37] to segment the current frame, and then the segmented superpixel blocks at multiple scales can be expressed as follows:

$$\{SP_1, SP_2, \ldots, SP_K\} = \{SP_{ij} | i \in [1, K], j \in [1, N_i]\},$$

where $i$ is the scale subscript used in multiscale segmentation, $K$ is the number of multiscales used, $N_i$ is the number of superpixels under scale $i$, and $j$ is the corresponding index of the superpixel block under a particular scale.

3.1.2. Foreground Detection. In foreground detection, the feature that distinguishes foreground and background is the RGB color feature. We use improved frame difference (IFD) to compute the difference between two corresponding superpixel patches in the current and background images. The IFD approach is not to calculate the feature values pixel by pixel but to perform a subtraction operation with a region of $R$ around a pixel and find the minimum value of the result as the variation of the pixel. This measure can effectively improve the robustness of the background subtraction algorithm and cope with the foreground detection of complex scenes such as dynamic backgrounds. IFD can be defined using the following formula:

![Figure 3: The pipeline of the proposed method SSSFD.](image-url)
where \((x, y)\) is the coordinate of a pixel point in the superpixel \((i,j)\). \(I_t\) represents the pixel value of the current frame, \((m,n)\) is the coordinate of pixels in the neighbor region of \((x,y)\), and \(R\) is the radius of the neighbor region.

We use a superpixel instead of a single-pixel for background subtraction. The superpixel, including the similarity and spatial relationship between pixels, can improve the performance of the proposed unsupervised algorithm. The foreground image consists of superpixels with foreground or background labels. Whether each superpixel is marked as foreground or background is determined by the variation in the RGB color characteristics of the superpixel itself. The color feature variation of a superpixel can be calculated by averaging the color feature changes of each pixel that composes the superpixel. This average, in turn, acts as a feature variation value for each pixel in the superpixel. Formula (3) can represent the process:

\[
P(SP_{i,j}) = M_{SP_{i,j}} \frac{\sum_{x,y \in SP_{i,j}} D(x,y)}{|SP_{i,j}|},
\]

where \(D(x,y)\) is obtained from formula (2), and represents the minimum difference of the feature values between the current frame and the background model in coordinate \((x,y)\). \(|SP_{i,j}|\) represents the number of pixels in the superpixel block.

After calculating the average value of superpixel, \(M_{SP_{i,j}}\) is compared with a threshold \(T_{sp}\) to determine whether the superpixel is foreground or background according to formula (4), so as to obtain the foreground segmentation map at a certain scale.

\[
Fg_i(x,y) = \begin{cases} 
1, & P(SP_{i,j}) > T_{sp}, \\
0, & \text{otherwise},
\end{cases}
\]

where \(Fg_i(x,y)\) means the foreground of the pixel with coordinate \((x,y)\) at scale \(i\), and \(T_{sp}\) is the threshold to judge whether a superpixel belongs to the foreground or not.

The number of superpixels in an image will affect the results of foreground detection. We use a multiscale strategy to effectively reduce the performance degradation of the algorithm caused by the uncertainty of the number of superpixels and better detect foreground objects. The formula is as follows:

\[
Fg(x, y) = \frac{\sum_{i=0}^{K} Fg_i(x, y)}{K},
\]

where \(Fg(x, y)\) is the foreground probability of the image, and \(K\) is the total number of used scales.

Finally, we compare the foreground probability map with a threshold to obtain the foreground detection result of the image. The formula is as follows:

\[
\text{Result} = \begin{cases} 
255, & Fg(x,y) > T_{fg}, \\
0, & \text{otherwise},
\end{cases}
\]

where \(T_{fg}\) is the threshold for judging the foreground.

3.1.3. Postprocessing. We use a region-based strategy when using the IFD to calculate the pixel difference between the current frame and the background model. This strategy will lead to the detection result being always slightly larger than the ground truth, called Region of Moving Object Slightly Larger (RMO-SL), as illustrated in [38]. Therefore, we use contour-optimizer (CO) [38] to optimize the background subtraction results to gain the final foreground segmentation map of the unsupervised method. The CO postprocessing method consists of suppression region, edge, and wrong edge detection. Suppression region detection aims to determine areas where foreground moving objects must not appear. Edge detection can generate the edges of the result. The wrong edges are detected by combing suppression regions and caught edges. After eliminating the wrong edges, we get the last outcomes of the MSS \(BS_t\), which are shown in the second column of Figure 4.
3.2. Train the Network for Semantic Segmentation. We adopt an advanced semantic segmentation network called DeepLabV3+ [23] and pretrained on the ADE20K dataset [39] to acquire initial weight parameters. ADE20K can be used for scene perception, parsing, segmentation, multijob recognition, and semantic understanding, with 2021/2000 densely annotated images for training and validation. There are 150 common objects in the dataset, denoted as $C = \{c_1, c_2, c_3, \ldots, c_{150}\}$. Following the same procedure as [40], we divide these objects into the foreground and background classes. The foreground class is $F_c = \{\text{person, car, cushion, box, book, boat, truck, bottle, van, bag, bicycle}\}$, and the background class is $B_c$, $B_c \cap F_c = \emptyset$, $B_c \subset C$.

Since there is no unified training dataset for foreground detection, we generally choose a certain number ($N$) of frames from CDNet2014 to form a training set when training a semantic segmentation network. There are usually three strategies for selecting frames [17]. One is to select one frame per $M/N$, where $M$ is the total number of video frames. The second is to choose $N$ frames randomly from the video. The third is to choose $N$ frames manually. Considering that the video contains many redundant frames, it will cause the problems of insufficient foreground information and an unbalanced number of positive and negative samples, which will affect the training effect of the network. Therefore, we use the manual selection of frames and select frames that contain foreground objects as much as possible for training, which will improve the accuracy of the network for foreground segmentation. Considering that there are a considerable amount of redundant frames in the video, actively screening valid frames instead of randomly selecting the training data can not only effectively reduce the amount of redundancy but also reduce the severe imbalance problem of positive and negative samples. We employ the dataset Sub-CDNet2014 provided by [41], which consists of 200 frames manually selected from each category of CDNet2014. The Sub-CDNet2014 is a total of 10,600 images, of which 80% are used for training, and the remaining 20% are used for validation. Transfer the pretrained network to Sub-CDNet2014.

Before training, to further improve the performance of the network, we perform data augmentation and positive and negative sample weight calculations on the images in Sub-CDNet2014. Data augmentation adopts standard geometric transformation methods such as inversion, translation. The weights of positive and negative samples are calculated as follows: expand every input ground truth into a corresponding one-dimensional vector, denoted as $P_1, P_2, \ldots, P_N$. $N$ is the number of input ground truth. Calculate the total number of pixels in $P$:

$$\text{Total}_P = \sum_{i=1}^{N} \sum_{t=1}^{L} P_i(t),$$

where $L$ represents the number of pixels in an image, and $t$ is the time sequence. Compute the classes of distinct pixel values in $P$:

$$C_t = \text{unique}(P_t),$$

where “unique” means to find the types of different classes in the ground truth. Calculate the frequency of occurrence of each value in each class:

$$H_t(\text{idx}) = \text{bincount}(P_t),$$

where “idx” represents the class, and “bincount” is used to count the frequency of pixels in each class. Compute the weight value of each class according to the following formula:

$$\text{weight}(\text{idx}) = \frac{\text{Total}_P}{(C_t \times H_t(\text{idx}))}.$$  

After the training is completed, the trained network is executed on the CDNet2014 dataset to obtain each image’s semantic foreground probability map. The probability is mapped to $SS_t$ in the range $0–255$ according to formula (11):

$$SS_t(x) = 255^* \frac{1}{\text{map}_t(x)},$$

where $x$ represents a fixed location of a pixel, and $\text{map}_t(x)$ is the image’s semantic foreground probability map.

The third column of Figure 4 shows some semantic foreground segmentation results.

3.3. Collaboration Process between Superpixel and Semantic Segmentation

3.3.1. Temporal Analysis of Semantic Background Model. In the semantic segmentation result, $SS_t$, stationary semantic foreground such as cars in a parking lot will be incorrectly classified as the foreground. This problem can be addressed by a similar approach to [8], that is, establishing a semantic background model (SBM). $SS_t^{\text{FG}}$ and $SS_t^{\text{BG}}$ two information with high confidence in foreground and background, respectively, are derived from $SS_t$. $SS_t^{\text{FG}}$ can be used to help reliably detect foreground pixels. At the same time, $SS_t^{\text{BG}}$ provides necessary information for detecting background pixels. They are defined as follows:

$$SS_t^{\text{FG}}(x) = SS_t(x) - SBM_t(x),$$

$$SS_t^{\text{BG}}(x) = SS_t(x),$$

where $SBM_t(x)$ represents the semantic background model at the pixel $x$ at time $t$. At first, initialize $SBM_t(x)$ with the first frame. We then continuously update the semantic background model to adapt to scene variations using a conservative update strategy that alleviates the problem of slow and intermittently moving objects blending into the background model. Specifically, the foreground pixels FG of the final detection result DR are not updated, and the background pixels BG are updated according to the learning rate $\bar{\delta}$.

$$\begin{align*}
\text{SBM}_t(x) &= SS_t(x), \\
\text{SBM}_{t+1}(x) &= \text{SBM}_t(x), \text{if } DR_t(x) = \text{FG}, \\
\text{SBM}_{t+1}(x) &= (1 - \bar{\delta}) \times \text{SBM}_t(x) + \bar{\delta} \times SS_t(x), \text{if } DR_t(x) = \text{BG}.
\end{align*}$$  

(14)
Figure 5 shows the updating process of $SBM_t(x)$.

3.3.2. Combination Rules. Using the detection results of MSS and semantic segmentation, we can determine the class of a pixel according to the defined rules.

First, in $SS_{BG}^t$, pixels with a low semantic probability should be classified as background points, regardless of the background subtraction result $BS_t$. Depending on this rule, challenges such as dynamic backgrounds, illumination variations, and cast shadows that seriously affect the performance of unsupervised foreground detection algorithms can be solved, and false positives in such scenarios can be greatly reduced, as shown in the first and second rows of Figure 4.

$$DR_t(x) = 0, \quad \text{if } SS_{BG}^t(x) \leq Th_{BG},$$  \hspace{1cm} (15)

where $Th_{BG}$ is the decision threshold value of background pixels.

Second, in $SS_{FG}^t$, pixels with high-semantic probability should be classified as foreground points. This rule reduces holes in foreground objects due to occlusion or model absorption, which tends to be false negatives, as shown in the third row of Figure 4:

$$DR_t(x) = 255, \quad \text{if } SS_{FG}^t(x) \geq Th_{FG},$$  \hspace{1cm} (16)

where $Th_{FG}$ denotes the decision threshold value of foreground points.

Finally, if the value of a pixel does not meet the above two conditions, the semantic segmentation result cannot provide enough decisions to determine the classification of the pixel. At this time, we directly use the MSS result $BS_t$ as the pixel class:

$$DR_t(x) = BS_t(x).$$  \hspace{1cm} (17)

The complete collaboration process is summarized in Table 1.

In brief, this process can also be represented by the following formula:

$$DR_t(x) = \begin{cases} BG, & \text{if } SS_{BG}^t(x) \leq Th_{BG}, \\ FG, & \text{elseif } SS_{FG}^t(x) \geq Th_{FG}, \\ BS_t(x), & \text{otherwise.} \end{cases}$$  \hspace{1cm} (18)

3.4. Background Update. The background update is significant. A timely and accurate update can make the algorithm better cope with complex scene changes and improve the robustness and accuracy of the algorithm. We adopt different update strategies for the pixels detected as foreground or background.

If a pixel is detected as a background pixel during foreground detection, the pixel value of the corresponding pixel in the background model is directly replaced with the current pixel value. We use a statistical histogram model to update the foreground pixel if a pixel is detected as foreground. Figure 6 shows the histogram statistics of pixel values for a pixel within a certain period. The abscissa is the pixel’s gray value at different times, and the ordinate is the frequency of occurrence of the gray value in this period. We utilize the statistical histogram’s peak to replace the background model’s pixel at the same location as the foreground pixel of the current frame. The process is denoted as follows:

$$p_{bm} = \hat{M}(P), P = \{p_1, p_2, \ldots, p_t\},$$  \hspace{1cm} (19)

where $p_{bm}$ is the pixel value in the background model at the same position as the foreground pixel of the current frame, and $\hat{M}$ represents the mode operation that computes the max value in a set, and $P$ is the set of pixel values in the time sequence from 1 to $t$.

The detailed procedure of the SSSFD algorithm for detecting moving objects is summarized in Algorithm 1.

4. Experiments

4.1. Dataset and Evaluation Metrics. CDNet2014 [9] dataset is currently the most extensive video dataset with pixel-level labels. It consists of 53 real-world scenarios in 11 different categories. All the video sequences are captured by fixed
cameras. This paper runs our proposed algorithm on five categories of the CDNet2014 dataset, evaluates the algorithms’ performance, and compares them with other representative foreground detection methods.

During the quantitative evaluation, we choose recall, precision, and $F$-measure [42] to evaluate the performance of the proposed approach. The recall is the detection rate, indicating the percentage of pixels correctly classified as foreground than foreground pixels in the ground truth. Precision is the accuracy rate, meaning the proportion of detected true positives to the number of all detected positives. $F$-measure is the weighted harmonic average of recall and precision, which is the comprehensive evaluation metric that can best reflect the performance of the foreground detection algorithm. The higher values of recall, precision, and $F$-measure, the better performance of the algorithm is. They are defined as follows:

Recall $= \frac{TP}{TP + FN}$

Precision $= \frac{TP}{TP + FP}$

$F$ - measure $= 2 \times \frac{\text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$  \hspace{1cm} (20)

where TP, FP, TN, and FN are true positive, false positive, true negative, and false negatives, respectively.

4.2. Experimental Results and Analysis. This section executes several experiments on the CDNet2014 dataset to analyze the proposed SSSFD algorithm in detail. We conducted an ablation study of SSSFD and performed qualitative and quantitative comparisons with other fundamental and state-

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{The background update of the foreground pixels in foreground detection.}
\end{figure}

\begin{table}[h]
\centering
\small
\begin{tabular}{cccccccc}
\hline
$j$ & $i$ & $R$ & $T_{sp}$ & $T_{fg}$ & $Th_{BG}$ & $Th_{FG}$ & $\vartheta$ \\
\hline
(8, 64, 256, 1024) & 4 & 2 & 0.3 & OTSU [43] & 200 & 225 & 0.00024 \\
\hline
\end{tabular}
\caption{The setting of hyperparameters.}
\end{table}
Figure 7: Visual results for MSS and SSSFD.

Figure 8: Performance comparison results of SSSFD and MSS in baseline, bad weather, dynamic background, shadow, and camera jitter video sequences.
of-the-art foreground detection algorithms to evaluate SSSFD. In this work, all experiments are implemented on a PC with a 3.6 GHz CPU, 16G RAM, and NVIDIA GeForce RTX 2080 Ti GPU in the Matlab 2020a environment under Windows 10.

A few hyperparameters should be set before initializing a background model, as shown in Table 2.

4.2.1. Ablation Study for SSSFD. Figure 7 presents the visual image comparison of MSS and SSSFD, and Figure 8 reports the quantitative comparison results of SSSFD and MSS in some scene categories of the CDNet2014 dataset. From the chart, we can note that the SSSFD algorithm that combines superpixel and semantic segmentation outperforms the superpixel-only foreground detection algorithm MSS in all categories.

SSSFD improves the $F$-measure by a significant boundary of 12.5% on the baseline category. This is mainly because the semantic segmentation network can completely segment potential foreground objects encountered in the training process. It is given a lift to $F$-measure by improving the foreground detection rate and more correctly recalled foreground pixels. SSSFD provides around 11.4% improvement compared to MSS in the bad weather category. MSS may be wrongly detected large snowflakes as foreground. Semantic segmentation can eliminate this false detection, reduce false positives, and improve precision and $F$-measure. Since MSS exploits the RGB color feature, it is easy to detect moving shadows as foreground. The high-performance semantic segmentation network can more accurately determine the shadow position through the extracted global information, eliminating the shadow.

Table 3: Comparison of detection results of SSN and SSSFD.

| Category              | Metrics   | SSN | SSSFD | Trend | SSN | SSSFD | Trend | SSN | SSSFD | Trend |
|-----------------------|-----------|-----|-------|-------|-----|-------|-------|-----|-------|-------|
| Baseline              | Recall    | 0.9668 | 0.9738 | ↑0.72% | 0.9805 | 0.9900 | ↑0.97% | 0.9736 | 0.9818 | ↑0.84% |
| Bad Weather           | Recall    | 0.9649 | 0.9718 | ↑0.72% | 0.9698 | 0.9768 | ↑0.72% | 0.9673 | 0.9743 | ↑0.72% |
| Dynamic Background    | Recall    | 0.9408 | 0.9695 | ↑3.05% | 0.9645 | 0.9817 | ↑1.78% | 0.9525 | 0.9756 | ↑2.42% |
| Shadow                | Recall    | 0.9776 | 0.9839 | ↑0.64% | 0.9838 | 0.9841 | ↑0.03% | 0.9807 | 0.9840 | ↑0.34% |
| Camera Jitter         | Recall    | 0.9715 | 0.9795 | ↑0.82% | 0.9753 | 0.9843 | ↑0.92% | 0.9734 | 0.9819 | ↑0.87% |

Figure 9: Qualitative results of the proposed method SSSFD and several state-of-the-art methods on CDNet2014. All the results are for the CDNet2014 dataset.
SSSFD also has a remarkable improvement relative to MSS in the shadow category, around 19.9%.

Dynamic background category is the different kind of challenge as camera jitter. There is the movement of some background objects in the dynamic background, such as curtains moving with the wind, swaying leaves, and sparkling water. The camera jitter video background contains no moving objects. The position of the entire camera changes irregularly due to external forces (such as wind blowing, vibration), resulting in the blurred video. Camera jitter will cause the pixels on the object’s edge to be detected as foreground points in the previous frame, the current frame as background points, and the next frame may be detected as foreground points. In the “fountain01” video sequence with dynamic background, the MSS algorithm detects the gushing fountain as the foreground while missing small natural moving objects, making this sequence detection fail. When the SSSFD is adopted, the false foregrounds are eliminated, and the correct ones are recalled. The F-measure has a surprising improvement on this sequence. The improvement in F-measure across the dynamic background category is 8.3%. Although MSS can recall most of the foreground moving object pixels in the camera jitter category, some false foreground pixels are also identified simultaneously due to drastic camera jitter, resulting in low precision of recalled pixels. However, SSSFD will segment the correct foreground object for each input frame, ignoring camera shake. Compared with MSS, the F-measure value of SSSFD is improved by an astonishing 80.9%.

The MSS is removed from SSSFD, and the quantitative foreground results obtained by the semantic segmentation network (SSN) and SSSFD are compared, as shown in Table 3. Each metric of SSSFD is better than SSN, which proves the role of MSS in SSSFD. The pixels that the semantic segmentation is not sure whether it is background or foreground, can be determined the class by MSS.

4.2.2. Comparison with Other Algorithms. This section presents the exhaustive evaluation of SSSFD. The proposed algorithm is compared with eleven fundamental and state-of-the-art approaches, including unsupervised methods like GMM [44], KDE [11], ViBe [12], PAWCS [45], SuBSENSE...
It is apparent from Figure 9 that SSSFD presents the best visual results, while GraphMOS and Cascaded CNN show competitive performance for these sequences. In the severe weather conditions of the "blizzard," only SSSFD and Cascaded CNN detect six cars in far-end motion, and other algorithms lost foreground objects more or less. The sequence "fall" presents dynamic background variations challenge because of a large area of swaying leaves. There are many false positives in GMM’s result because a single statistical model cannot describe the complex dynamic background. GraphMOS detects a stationary car in a parking lot as the foreground object, which may happen in some supervised algorithms. Benefiting from the supplementary information from the semantic background model, SSSFD effectively classifies stationary cars as the background. The "overpass" is another typical dynamic background sequence that contains a little shadow. Only SSSFD and GraphMOS can accurately locate pedestrians. However, due to the over-smoothing of GgraphMOS, the pedestrian is visually slightly rounder than the ground truth. In the "peopleInShade" and "badminton" sequences, a similar situation also occurs. The "sidewalk" is a video shot by a camera shakes violently. The camera jitters cause the position of all objects to move throughout the image, which is a massive challenge for any foreground detection algorithm. We observe that GMM cannot detect pedestrians because the variations of pixels in the picture do not follow a Gaussian distribution. The foreground masks provided by SuBSENSE and IUTIS-5 are also an incomplete part of the pedestrian's body. The detection results of SSSFD also lose part of the foreground, which is caused by the inability of the semantic segmentation network to segment small objects with irregular shapes accurately.

Table 4 proves that the performance of the deep learning-based foreground detection algorithm is much higher than that of the traditional unsupervised algorithm. Although it is controversial whether supervised foreground detection algorithms can be applied in actual video surveillance, we believe that high-performance supervised algorithms are helpful for some monitoring situations where foreground objects are relatively fixed. SSSFD achieves the best results in four out of five challenges, including baseline (0.9818), bad weather (0.9743), dynamic background (0.9756), and camera jitter (0.9819), while it gains a little shadow (0.9840).

In conclusion, it is evident from the qualitative and quantitative results that SSSFD obtains better performance and is closer to the ground truth in most video sequences.

5. Conclusion

This paper proposes a foreground detection algorithm called SSSFD. It utilizes an unsupervised algorithm based on multiscale superpixel segmentation to acquire background subtraction results. We train a semantic segmentation network on a subset of manually selected frames from CDNet2014 and employ this network to perform semantic segmentation on input images. Updating the background model with fine foreground results composed of superpixel and semantic segmentation dramatically improves the accuracy of foreground detection. The experimental results on the CDNet2014 dataset verify the effectiveness of the proposed algorithm. SSSFD achieves state-of-the-art performance in visual and qualitative evaluation compared with other algorithms.

The limitation of the SSSFD algorithm is that the real-time performance is poor. This is because, first, multiscale superpixel segmentation takes up a lot of computational costs. Second, the processing speed of the high-precision semantic segmentation network is slow. Therefore, our future work will investigate more accurate and faster semantic segmentation networks and unsupervised background subtraction algorithms, as well as efficient fusion strategies for semantic information and background subtraction results.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2022-2018-08-01417) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

[1] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection - a systematic survey," Recent Patents on Computer Sciences, vol. 4, no. 3, pp. 147–176, 2011.
[2] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: an overview," Computer science review, vol. 11-12, pp. 31-66, 2014.
[3] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," Neural Networks, vol. 117, pp. 8–66, 2019.
[4] J. H. Giraldo, H. T. Le, and T. Bouwmans, "Deep learning based background subtraction: a systematic survey," in Handbook of Pattern Recognition and Computer Vision, pp. 51–73, World Scientific Publishing, Singapore, 2020.
[5] X. Ren and J. Malik, "Learning a classification model for segmentation," in Proceedings of the 9th IEEE International Conference on Computer Vision, vol. 2, p. 10, IEEE Computer Society, Nice, France, October 2003.

[6] Z. Xu, B. Min, and R. C. Cheung, "A robust background initialization algorithm with superpixel motion detection," Signal Processing: Image Communication, vol. 71, pp. 1–12, 2019.

[7] Y.-Q. Chen, Z.-L. Sun, and K.-M. Lam, "An effective sub-superpixel-based approach for background subtraction," IEEE Transactions on Industrial Electronics, vol. 67, no. 1, pp. 601–609, 2020.

[8] M. Braham, S. Piérard, and M. Van Droogenbroeck, "Semantic background subtraction," in Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), pp. 4552–4556, IEEE, Beijing, China, September 2017.

[9] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezheth, and P. Ishwar, “CDNet 2014: an expanded change detection benchmark dataset,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 387–394, Columbus, Ohio, USA, June 2014.

[10] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," vol. 2, pp. 246–252, in Proceedings of the 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149), vol. 2, pp. 246–252, IEEE, Fort Collins, CO, USA, June 1999.

[11] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," Proceedings of the IEEE, vol. 90, no. 7, pp. 1151–1163, 2002.

[12] O. Van Droogenbroeck and M. Van Droogenbroeck, "ViBe: a universal background subtraction algorithm for video sequences," IEEE Transactions on Image Processing, vol. 20, no. 6, pp. 1709–1724, 2011.

[13] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: a universal change detection method with local adaptive sensitivity," IEEE Transactions on Image Processing, vol. 24, no. 1, pp. 359–373, 2015.

[14] S. Lu and X. Lu, "WeSamBE: a weight-sample-based method for background subtraction," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2105–2115, 2018.

[15] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in Proceedings of the 2016 international conference on systems, signals and image processing (IWSSIP), pp. 1–4, IEEE, Bratislava, Slovakia, 23–25 May 2016.

[16] C. Lin, B. Yan, and W. Tan, "Foreground detection in surveillance video with fully convolutional semantic network," in Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 4118–4122, IEEE, Athens, Greece, 07-10 October 2018.

[17] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," Pattern Recognition Letters, vol. 96, pp. 66–75, 2017/09/01/2017.

[18] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, Boston, MA, USA, June 2015.

[19] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2881–2890, Honolulu, Hawaii, USA, July 2017.

[20] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected Crfs," 2014, https://arxiv.org/abs/1412.7062?context=cs.

[21] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, 2018.

[22] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Re-thinking atrous convolution for semantic image segmentation," 2017, http://arXiv.org/abs/055387.

[23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in Proceedings of the European conference on computer vision (ECCV), pp. 801–818, Munich, Germany, September 2018.

[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, 2017.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arXiv.org/abs/1409.1556.

[26] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puig, and Y. Ruichek, "BSCGAN: deep background subtraction with conditional generative adversarial networks," in Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 4018–4022, IEEE, Athens, Greece, 07-10 October 2018.

[27] W. Zheng and K. Wang, "Background subtraction algorithm with bayesian generative adversarial networks," Zidonghua Xuebao/Acta Automatica Sinica, vol. 44, pp. 05–27, 2018.

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

[29] M. Babaei, D. T. Dinh, and G. Rigoll, "A Deep Convolutional Neural Network for Background Subtraction," 2017, http://arXiv.org/abs/01731.

[30] X. Zhao, Y. Chen, M. Tang, and J. Wang, "Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network," in Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), pp. 343–348, IEEE, Hong Kong, China, July 2017.

[31] X. Li, M. Ye, Y. Liu, and C. Zhu, "Adaptive deep convolutional neural networks for scene-specific object detection," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2538–2551, 2019.

[32] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise deep sequence learning for moving object detection," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 9, pp. 2567–2579, 2019.

[33] K. Lim, W.-D. Jang, and C.-S. Kim, "Background subtraction using encoder-decoder structured convolutional neural network," in Proceedings of the 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), pp. 1–6, IEEE, Lecce, Italy, August 2017.

[34] Z. Hu, T. Turki, N. Phan, and J. T. L. Wang, "A 3D atrous convolutional long short-term memory network for background subtraction," IEEE Access, vol. 6, pp. 43450–43459, 2018.

[35] D. Sakkos, H. Liu, J. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural
