IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INTEGRATIVE COMPUTER VISION AND MULTIMEDIA ANALYTICS

Received September 12, 2019, accepted October 23, 2019, date of publication October 31, 2019,
date of current version November 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2950639

Universal Foreground Segmentation Based on Deep Feature Fusion Network for Multi-Scene Videos

YE TAO[1], ZHIHAO LING[1], AND IOANNIS PATRAS[2], (Member, IEEE)
[1]Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K.

Corresponding author: Zhihao Ling (zhhling0425@163.com)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 222201917006.

ABSTRACT Foreground/background (fg/bg) classification is an important first step for several video analysis tasks such as people counting, activity recognition and anomaly detection. As is the case for several other Computer Vision problems, the advent of deep Convolutional Neural Network (CNN) methods has led to major improvements in this field. However, despite their success, CNN-based methods have difficulties in coping with multi-scene videos where the scenes change multiple times along the time sequence. In this paper, we propose a deep features fusion network based foreground segmentation method (DFFnetSeg), which is both robust to scene changes and unseen scenes comparing with competitive state-of-the-art methods. In the heart of DFFnetSeg lies a fusion network that takes as input deep features extracted from a current frame, a previous frame, and a reference frame and produces as output a segmentation mask into background and foreground objects. We show the advantages of using a fusion network and the three frames group in dealing with the unseen scene and *bootstrap* challenge. In addition, we show that a simple reference frame updating strategy enables DFFnetSeg to be robust to sudden scene changes inside video sequences and prepare a motion map based post-processing method which further reduces false positives. Experimental results on the test dataset generated from CDnet2014 and Lasiesta demonstrate the advantages of the DFFnetSeg method.

INDEX TERMS Convolutional neural network, foreground segmentation, multi-scene videos aware.

I. INTRODUCTION

Foreground segmentation, also named as fg/bg classification, that is the segmentation of frames into the background and foreground pixels is a commonly used first step for detecting regions of interest in videos, which has the same effect as the well-known task background subtraction but with a different mechanism. Foreground extraction helps video analysis methods to discard irrelevant information in applications such as video surveillance [1], pose estimation [2], and face detection [3].

Traditionally, fg/bg classification methods mainly focus on static surveillance camera videos, where the background pixels depict either static regions or regions with semi-periodic motion (e.g. flowing water). However, the development of camera hardware enables the surveillance camera to be portable, which brings the challenge of scene change to the surveillance videos caused by the camera position or location change. We name these videos as multi-scene surveillance videos. The multi-scene challenge is not new for the fg/bg classification task, as the *twoPositionPTZCam* video in the largest change detection algorithm benchmark dataset — CDnet2014 [4] is a multi-scene video. Many fg/bg classification methods [5], [6] are designed based on the assumption that the camera is static, but recent methods begin to pay attention to non-static scenes. For instance, the special components to deal with the scene change problem are proposed in many traditional methods such as SubSENSE [7], SWCD [8] and so on. Also, a flux tensor [9] based scene change detection method is also used by a deep learning background subtraction method [10]. However, since their performance relies on the quality of the background model, they are still spoiled by the unstable background model caused by the scene change.

The associate editor coordinating the review of this manuscript and approving it for publication was Vladimir M. Mladenovic.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

On the other hand, recent supervised deep learning methods [11], [12] deal with the multi-scene problem by simply single frame foreground segmentation. They do not consider the background model and generate the foreground mask from every single frame, achieving near human-level performance, which surpasses the traditional methods with a large gap. Nevertheless, in the original paper, the parameters of these models are trained and tested on one specific video or a group of videos and the performance on unseen videos has not been evaluated. (Unseen videos denote that the videos whose scenes have no overlap with the scene in training dataset.) For example, FgSegNet [11] trains one model for one video with 200 frames and get a super good result when it tests the performance on the rest frames from the same video, whereas as shown in the experiment in Section IV-E, its performance drops significantly for unseen videos even with more than one hundred times various training data. As the real world environment is generally changeful and uncontrollable, it is not possible to guarantee that one model will only work on the known scene. Therefore, a universal foreground mask generation method which is robust to both unseen and multi-scene videos is necessary.

In this paper, we propose a deep feature fusion network based foreground segmentation method (DFFnetSeg) to tackle the problem mentioned above. DFFnetSeg takes as input a single frame, a previous frame and a reference frame, and produces in the output a segmentation mask for fg/bg separation. The reference and previous frames carry the long and short term information in one sequence which enables DFFnet not only to reserve the temporally stopped object mask but also to eliminate the *ghost* mask. In addition, our model is universal to a wide range of unseen videos (including videos from indoor, outdoor, under different weather, and so on), which works stably when both background scenes and foreground objects are totally unseen during training. A simple Pearson correlation coefficient based reference frame updating strategy further enables DFFnetSeg robust to the scene change inside the videos. As a reference frame is used instead of a background model, no extra effort is needed for background modelling, which leads to a fast response to the scene change.

The contributions of this paper are three folds:

- We propose a deep features fusion network which first compares features extracted from Pyramid scene parsing network (PSPNet) in different depth levels to generate soft motion maps and then fuses the various levels of soft motion maps and single frame feature maps to produce in the output the fg/bg segmentation mask. We show that with the help of semantic information extracted by the PSPNet, high-quality segmentation masks are achievable even without background modelling.
- We propose a new post-processing method based on region-level motion map, which eliminates the false positive classification so as to boost the foreground mask.
- We propose a simple Pearson correlation coefficient based reference frame updating strategy which is both effective and efficient.

The paper is structured as follows. In Section II, we will discuss the related works. In Section III, we will present the DFFnetSeg method in detail. In Section IV, we will describe the experiments and discuss the results. In Section V, we will conclude the DFFnetSeg method and discuss the future work.

II. RELATED WORK

As is the case in several Computer Vision tasks, fg/bg classification methods could be classified into two categories: traditional methods and deep learning methods – the latter appearing to dominate the field in the recent years.

The traditional methods generally go through the pipeline of background model construction, background model maintenance and subtraction. Classified by background models, Gaussian mixture model (GMM) based methods [9], [13], codebook-based methods [14] and sample model based methods [7], [15], [16] occupied the top place in terms of performance. Typical GMM-based methods fit a Gaussian mixture model as the probability density function to describe the colour/intensity/features distribution at each pixel. Recently, Wang *et al.* [9] combine the flux tensor based motion detection method with split Gaussian models in order to deal with more complex scene challenges, such as illumination changes and ghosting effects. Chen *et al.* [13] propose a sharable GMM model which extends its robustness to camera jitter and dynamic background challenge by developing the spatial-temporal correlation between pixels. Original codebook based methods describe each pixel by a codebook containing a set of codewords and each codeword represents a range of pixels' intensity values of the background, whereas the recent variant like PAWCS [14], which utilizes the colour/LBSP/persistence triplets to construct the robust background words model, and dynamically adjust thresholds and learning rates for segmentation decision and model updating rules. Different from other encoding models, sample-based models just sample values from previous frames in order to maintain a background model. In such methods, background models are updated based on updating probabilities that are estimated spatiotemporally and the subtraction part is implemented by comparing the number of matching samples in background models with a threshold. Their variants are further proposed to enhance the robustness for different challenges. For example, SubSENSE [7] adds spatiotemporal binary features intra-LBSP and inter-LBSP to the background model and comparison stage, and gets it updated adaptively by monitoring the model fidelity and local segmentation noise. Besides, WeSamBE [16] employs a weight mechanism to both the background model and updating. As the unsupervised methods, they are not limited to certain video and can obtain proper performance by default parameters. However, as is typically the case in Computer Vision task, the handcrafted features have difficulty in coping with more complex situations.

To address this shortcoming, the first deep background subtraction method [17] was proposed with convolutional neural network (CNN) in 2016 and since then, tons of deep learning fg/bg classification methods spring out and surpass the traditional methods with a great gap (around by 20% in terms of F-Measure). The original deep background subtraction method [17] simply uses a pre-calculated image as the background model for each sequence and employs a network similar to LeNet-5 for subtraction, which is evaluated in a scene-specific manner on selected sequence from CDnet2014 dataset with F-Measure 0.9. Based on the similar network architecture, DeepBS [10] updates the background image based on SubSENSE [7] and flux tensor [9] along the sequence, which is evaluated in a relatively universal manner with F-Measure 0.75. Follow a similar background image generation method as [17], generative adversarial networks (GANs) based methods [18]–[20] are proposed with F-Measure around 0.95. BScGAN [18] and BGAN [19] utilize the conditional GAN and Bayesian GAN respectively and BPVGAN [20] further introduces parallel vision to the BSGAN. Different from the methods which need to maintain a background model, the recurrent neural network based methods extend their scope to the temporal sequence. SFEN [21] first extracts semantic maps from a single frame as the input of a ConvLSTM, and an STN model [22] and CRF [23] are combined to enhance the motion robustness and spatial smoothness of the output mask. Hu *et al.* [24] conduct 3D atrous convolutional network with multi-frame input before ConvLSTM. By contrast to the sequence-based method, the foreground segmentation methods [11], [12] only consider a single frame to segment foreground. Among them, FgSegNet [11] and its variants occupy first several entries on CDnet2014 with f-measure around 0.98. FgSegNet generates the foreground segmentation mask by using a single frame as input to the encoder-decoder structure architecture which uses triplet VGG-16 Net as the encoder and a transposed convolutional neural network as the decoder. Different from all the deep learning methods mentioned above, SemanticBGS [25] combines deep learning semantic segmentation with traditional methods, without training, in which rules are made to combine semantic maps from pre-trained PSPNet [26] with the traditional methods without modifying their internal elements. As a result, SemanticBGS reduces the mean overall error rate of 34 traditional algorithms by roughly 50%.

Most deep learning methods surpass other methods in terms of the evaluation metrics, but their good performance is benefited from the background scene overlap and even foreground object overlap between training data and testing data. For example, Cascade CNN [12] manually chooses 200 frames from each video as the training set and SFEN [21] uses the first half of videos as the training set with the rest as testing ones. By contrary, SemanticBGS shows its advantage as an unsupervised method on universality which is not limited by the training and testing set. Also, the high-quality performance of SemanticBGS mainly benefits from the pre-trained deep semantic segmentation stage. It is because the deep semantic segmentation methods [26], [27] are trained on large and varied datasets which include enough semantic classes to be the reference for foreground segmentation. Therefore, dynamic background such as shaking trees and *ghost* mask on the road region caused by the removed car are easy to be eliminated, from the semantic perspective. Based on the superiority of deep semantic segmentation, the DFFnet internally combines the semantic information with deep learning method to extend the universality of deep learning methods and the robustness to challenging situations.

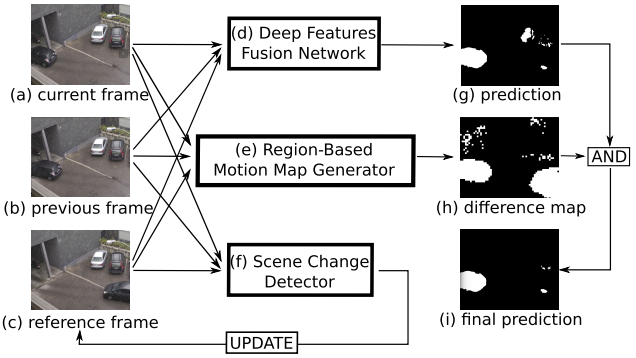


FIGURE 1. An illustration of the pipeline of DFFnetSeg method. Given a group of input images consist of a current frame (a), a previous frame (b) and a reference frame (c), we use three parts in parallel which consist of a deep features fusion network (d), a region-based motion map generator (e) and a scene change detector (f) to extract the foreground mask prediction (g), generate the motion map (h) and update the reference frame (c), respectively. Finally, we use the motion map (h) to boost the final prediction (i) by the bitwise and operator.

III. PROPOSED METHOD

Let us denote by $f^{(t)} \in \mathcal{R}^{w\times h\times 3}$, $t \in [1, T]$ the RGB frame at time t of an image sequence with T frames in all, where w and h are width and height. The DFFnetSeg method aims to produce a label mask $M^{(t)}_{post} \in \mathcal{R}^{w\times h}$ each entry of which denotes whether the corresponding pixel depicts a foreground object or the background, from the input group which is compound of the current frame $f^{(t)}$, the previous frame $f^{(t-4)}$ and the reference frame. The reference frame is initialized by $f^{(1)}$ and updated along the sequence.

The DFFnetSeg consists of three parts in parallel: a deep features fusion network, a region-based motion map generator and a scene change detector, as shown in Fig 1.

In detail, the first part is a convolutional neural network, which includes two stages: the deep semantic features extractor and the foreground mask generator based on feature fusion. In the features extractor stage, we feed each entry from the input group into a pre-trained PSPNet respectively to extract the deep features. In the mask generator stage, features of each entry are compared at selected depth levels. The inner group comparisons are further fused with the image content features extracted from $f^{(t)}$ at each depth level, as shown in Fig 2. Finally, the shallow feature maps and the deep feature maps are fused to generate the mask prediction $M^{(t)}$.

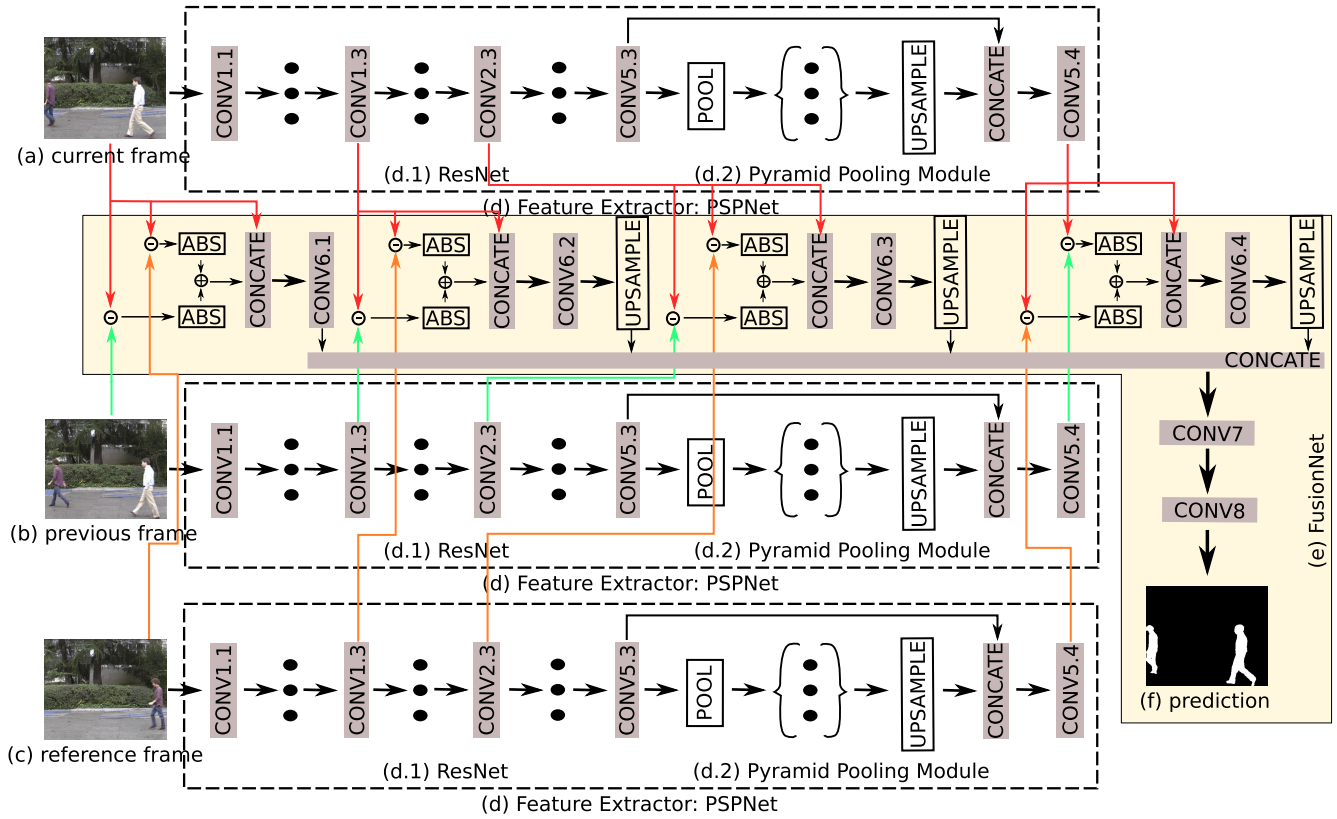


FIGURE 2. Architecture of our proposed DFFnet. Firstly, features from chosen layers of PSPNet (d), which consists of ResNet (d.1) and Pyramid Pooling Module (d.2), are extracted, given inputs including a current frame (a), a previous frame (b) and a reference frame (c), respectively. After that, for each chosen layer, a FusionNet (e) first obtains the sum of the difference of features between (a) and (b), and between (a) and (c), respectively. Then it concatenates the difference sum and features from (a) to form a presentation which carries both content information of (a) and motion information along frames, followed by a convolutional layer to combine this information and upsampling (optional) layers to normalize the output shape. Finally, the features from different levels are concatenated to form the final feature representation, followed by two convolutional layers to fuse local information with the global one and get the final per-pixel prediction (f).

The second part is a region-based motion map generator, which could be regarded as a post-processing step to reduce the false positives caused by semantic noise. Specifically, the false positives here are the pixels which are classified as foregrounds because they belong to some objects which have pretty high possibility to be foregrounds such as humans and cars but they are static. This part just simply conducts a region-level comparison between $f^{(t)}, f^{(t-4)}$ and the reference frame f_{ref} to get the motion map which indicates the potential motion region. The motion map is then used to post-process $M^{(t)}$ as the final prediction $M^{(t)}_{post}$.

The third part is a scene change detector, which utilizes the Pearson correlation coefficient to indicate the degree of scene change and update the reference frame when the large scale scene change is detected.

These three parts are discussed in detail as following.

A. NETWORK ARCHITECTURE

In the architecture of our DFFnet shown in Fig 2, the feature extractor and the fusion network are shown in white and cream-coloured background respectively.

1) FEATURE EXTRACTOR

SFEN [21] and SemanticBGS [25] show that semantic information plays an important role in fg/bg classification. Specifically, naive background subtraction is easy to meet the *ghost* problem when moving objects have not been eliminated from the background model. Even when the background model is super clean (without any foreground objects), the *camouflage* problem may happen when the foreground object shares a similar colour with the background region. However, *ghost* and *camouflage* problems are easy to be overcome if we know whether the problem region belongs to a potential moving object or not, from the semantic knowledge. Therefore, the semantic segmentation network comes to mind. Different from SFEN and SemanticBGS which only use final layers of a semantic segmentation network, the DFFnet uses both shallow layers and deep layers of a deep semantic segmentation network PSPNet. The shallower layers capture the low-level features such as edge, corner, and shape; the deeper layers capture the high-level features such as semantic information. Both of them contribute to the high-quality foreground mask generation.

TABLE 1. The details of the architecture of the ResNet part of PSPNet.

layer name	output size	layers info
conv1.1	237×237	3×3, 64, stride 2
conv1.2	237×237	3×3, 64, stride 1
conv1.3	237×237	3×3, 64, stride 1
conv2.x	119×119	3×3 max pool, stride 2 $\begin{Bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{Bmatrix} \times 3$
conv3.x	60×60	$\begin{Bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{Bmatrix} \times 4$
conv4.x	60×60	$\begin{Bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{Bmatrix} \times 6$
conv5.x	60×60	$\begin{Bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{Bmatrix} \times 3$

The PSPNet we used as the feature extractor is trained on ADE20K dataset [28], [29] because the ADE20K dataset includes various scenes and objects which have similar view angles as surveillance videos. In terms of the architecture, the PSPNet consists of the ResNet50 to extract feature maps and the pyramid pooling module to generate semantic segmentation maps. The ResNet50 is slightly different from the original one, whose details are shown in Table 1. In the pyramid pooling module, we use the average pooling with bin sizes of 1×1, 2×2, 3×3 and 6×6 respectively and the convolutional layer is 512 feature maps with 1×1 kernel (the detail architecture sees [26]). The final feature representation obtained by concatenation is followed by the 3×3 kernel, 512 maps convolutional layer CONV5.4. Batch normalizations are applied after each convolutional layer and activation function is ReLU.

2) FUSION NETWORK

The fusion network conducts two kinds of fusion. Firstly, considering each feature level, the fusion network fuses image contents with the difference between input entries. Secondly, considering across feature levels, the fusion network fuses local information with global one and low-level features with high-level features.

Specifically, the current frame $f^{(t)}$, the previous frame $f^{(t-4)}$ and the reference frame are fed to PSPNet respectively to extract the corresponding features. The last layers of each scale of PSPNet are chosen as the input of the fusion network, which are CONV1.3, CONV2.3, and CONV5.4 respectively. The RGB images are also regarded as a grouped entry of the fusion network. Thus, the fusion network considers four-level features in all. These four-level features include both four scale levels which contain various degree of local and global information and four depth levels which contain various degree of shape and semantic information.

Let us denote by $A_l(f^{(t)})$, $l \in \{1, 2, 3, 4\}$ the features at level l of input $f^{(t)}$. We define the soft motion map at level l, denoted by $B_l = (d^l_{ijk})$, given denotion $A_l(f^{(t)}) = (a^l_{ijk})$, $A_l(f^{(t-4)}) = (b^l_{ijk})$ and $A_l(f_{ref}) = (c^l_{ijk})$, as follows:

$$d^l_{ijk} = \left|a^l_{ijk} - b^l_{ijk}\right| + \left|a^l_{ijk} - c^l_{ijk}\right|, \tag{1}$$

where i, j, k denote the index of one element in each dimension. d^l_{ijk} is relatively high when the features of current frame $f^{(t)}$ are different from those of both the previous frame and the reference frame, which denotes the location (i,j) may contain motion. The soft motion map can also be regarded as the external comparison of current frame information with short term information and long term information. The previous frame $f^{(t-4)}$ which only has short time interval with the current frame carries the short term information, while the reference frame carries the long term information because it only updates when a large scene change is detected. The advantage of short term information for foreground object detection is the robustness to the continuous changing background because closer frames may share a more similar background than others. However, when the foreground object temporarily stops for a while, the foreground object may be lost if only considering short term information. Then, the long term information becomes an important reference.

The soft motion maps are concatenated with the features of $f^{(t)}$ to feed to the CONV6.x layers to fuse motion information with frame contents. All the CONV6.x layers generate 32 feature maps with 3×3 kernel size. For level 2 to 4, the spatial upsampling is used to normalize the output back to the same size as level 1.

The feature maps from 4 levels are further concatenated and then fed to the CONV7 to fuse both low-level features with semantic information and local with global information, which generates 32 feature maps with 3×3 kernel size. Finally, the CONV8 uses 1×1 convolutional layer with softmax to produce the foreground object mask $M^{(t)} \in \{0, 1\}^{w \times h}$ (foreground $= 1$, background $= 0$).

To optimize the weights in the fusion network, the corresponding loss is then the cross-entropy loss function, that is,

$$Loss = -\frac{1}{I}\sum_{i=1}^{I}[C(i)\log p(i) + (1 - C(i))\log(1 - p(i))], \tag{2}$$

where $C(i)$ is the ground truth label and $p(i)$ is the output of the network at pixel location i.

B. REGION-BASED MOTION MAP

Different from the soft motion map in the fusion network, this part proposes a hard motion map which gives the binary value to each region to denote motion by comparing the difference between $f^{(t)}$, $f^{(t-4)}$ and f_{ref}.

More specifically, we first define the motion mask $D_{pre} = (p_{ij})$ and $D_{ref} = (r_{ij})$, given denotions $f^{(t)} = (a_{ij})$, $f^{(t-4)} = (b_{ij})$ and $f_{ref} = (c_{ij})$, as follows:

$$p_{ij} = \begin{cases} 1 & \text{if } (|a_{ij} - b_{ij}|) > \theta \\ 0 & \text{otherwise}, \end{cases}$$

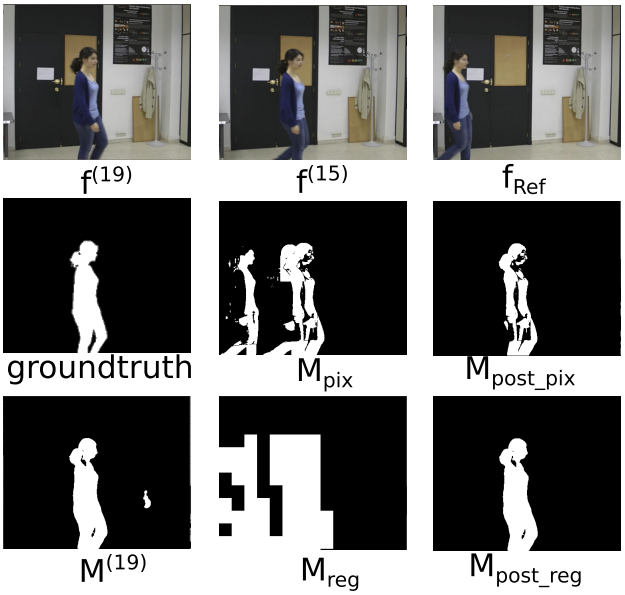


FIGURE 3. An example of the motion map based post-processing. The current frame $f^{(19)}$, the previous frame $f^{(15)}$ and the reference frame f_{ref} are the 19th, 15th and 1st frames of the selected clip of *I_SI_01* sequence of LASIESTA dataset respectively. $M^{(19)}$ is the raw foreground mask estimated by the fusion network. M_{pix} and M_{reg} are the pixel level motion mask (with $\theta = 20$) and region-based motion map (with $\theta = 20$, $\beta = 5$ and $N = 32$) respectively. M_{post_pix} and M_{post_reg} are the post-processed $M^{(19)}$ by M_{pix} and M_{reg} respectively.

$$r_{ij} = \begin{cases} 1 & \text{if } (|a_{ij} - c_{ij}|) > \theta \\ 0 & \text{otherwise} , \end{cases} \tag{3}$$

where i and j denote the location of pixels. A pixel is regarded as moving when the difference is larger than θ. Notice, here we only consider the greyscale of images.

The aim for hard motion map is to clean the false positives of $M^{(t)}$ caused by potential moving objects. Therefore, we propose it as a greedy mask, which activates one entry even with a relatively weak hint for motion. In detail, the bitwise or operator is used to obtain the pixel level motion mask $M_{pix} = (m_{ij}^{pix})$ as follows:

$$m_{ij}^{pix} = \begin{cases} 1 & \text{if } p_{ij} = 1 \text{ or } r_{ij} = 1 \\ 0 & \text{otherwise} \end{cases} . \tag{4}$$

Next, the pixel level motion mask is further transferred to the region-based motion map which can reduce the effect of local *camouflage* problem (an example is shown in Fig 3). We divide the whole motion mask to $N \times N$ regions without overlapping (the edge is padded to be valid), denoted by $\mathbf{m}_k \subseteq M_{pix}$ (with $\bigcup_k \mathbf{m}_k = M_{pix}$). Then, the region-based map $M_{reg} = (m_{ij}^{reg})$ is obtained based on the quantity of motion in each region as follows:

$$m_{\Psi_i}^{reg} = \begin{cases} 1 & \text{if } \sum_{(i,j)\in\Psi_i} m_{ij}^{pix} > \beta \\ 0 & \text{otherwise} , \end{cases}$$
$$\text{with } \Psi_i = \{(i,j) | m_{ij}^{pix} \in \mathbf{m}_k\}. \tag{5}$$

The region entry of map M_{reg} is activated when the quantity of motion in the specific region is larger than β.

The region-based map is finally used to post-process the estimated mask generated by the fusion network. Denoted $M_{post} = (m_{ij}^{post})$ is defined by:

$$m_{ij}^{post} = \begin{cases} 1 & \text{if } m_{ij}^{reg} = 1 \text{ and } m_{ij} = 1 \\ 0 & \text{otherwise} . \end{cases} \tag{6}$$

where $M = (m_{ij})$ denotes the output of the fusion network. The bitwise and operator is used here to restrict the final prediction. As shown in Fig 3, the pixel level motion mask can not detect sufficient motion pixels because of camouflage phenomenon, which leads to the holes inside the foreground object in final prediction, whereas the region level motion mask can tackle this well. The *ghost* mask exists in both the pixel level mask and the region-based map because they are defined greedily but the *ghost* has little effect on final prediction. In addition, as we can see in $M^{(19)}$, there is a small region of false positives, which is misclassified by the fusion network because it corresponds to the *clothes* region in the frame which has a high probability to be a moving object (as it is usually carried by human), but it is easy to be eliminated by the motion map.

C. SCENE CHANGE DETECTOR

If the reference frame in the input group of the network does not change along time, it will lose its advantage and even conduct the opposite effect when the dramatic background change caused by the camera position or location change happens. To tackle this problem, a simple Pearson correlation coefficient based scene change detector is proposed to decide whether to update the reference frame or not at time t.

The Pearson correlation coefficient is defined as (7), where $\forall x, y \in R^{n\times 1}$ and $n \in N_+$.

$$Col(x, y) = \frac{(x - \bar{x})^T (y - \bar{y})}{\|(x - \bar{x})\|_2 \|(y - \bar{y})\|_2} . \tag{7}$$

Then, $Col_1^t = Col(Ds(f^{(t)}), Ds(f^{(t-4)}))$ and $Col_2^t = Col(Ds(f^{(t)}), Ds(f_{ref}))$ are the correlation coefficient between the current frame, previous frame and reference frame respectively, where $Ds(\cdot)$ denotes spatial downsampling image to 32×32 size and flattening it. The 32×32 size can both reduce the computational load and capture enough information to denote a scene. Intuitionally, the correlation coefficient can denote the similarity between two images. Therefore, based on the coefficient, the reference frame updating strategy flow is as follows:

- We define an indicator $C_t \in \{0, 1\}$ which is 1 when $Col_1^t < \gamma$ and $Col_2^t < \gamma$ at time t, because generally the correlation is regarded as weak enough to judge that the scenes are different when the coefficient is lower than γ.
- We update the reference frame by $f_{ref} = f^{(t-4)}$ and reset $C_j = 0$ ($j \in [t-4, t]$) only when three requirements are fulfilled: a) $C_{t-4} = 1$, b) $\sum_{i=1}^{3} C_{t-i} > 1$ and c) $Col_2^t > \gamma$. It first detects a potential scene change

signal, then further detects the scene change in neighbour frames to tolerate the potential noisy detection, and finally, update the reference frame by the farthest same scene frame to guarantee the long term information.

Updating the reference frame when the scene changes enables the DFFnetSeg to be robust to deal with the multi-scene videos caught by multi-position or multi-location cameras.

IV. EXPERIMENTS AND RESULT

A. DATASET

We evaluate the performance of DFFnetSeg method by two datasets CDnet2014 [4] and LASIESTA [30], to sufficiently test the universality of DFFnetSeg.

The CDnet2014 dataset is the largest change detection benchmark and dataset, including the evaluation matrics, the rank of state-of-the-art methods and the pixel-level ground truth of 53 sequences. These sequences from different scenes are separated to 11 challenge categories, including bad weather (BW), baseline (Ba), camera jitter (CJ), dynamic background (DB), intermittent object motion (IOM), low frame rate (LF), night video (NV), shadow (Sh), thermal camera (TH), air turbulence (TB), and pan-tilt-zoom camera (PTZ), and each category includes 4 to 6 sequences. In our experiment, we do not consider the continue camera moving and air turbulence videos because our DFFnetSeg method focuses on the multi-scene videos captured by the intermittent position changed camera, and the air turbulence is out of the scope of the scene domain considered in this paper. Therefore, we only include the "twoPositionPTZCam" sequence in the PTZ category and exclude all the sequences in the TB category.

In addition, 7 objective evaluation matrics provided by CDnet2014 are used to evaluate the performance of algorithms quantitatively:

- Re (Recall) : TP / (TP + FN).
- Sp (Specificity) : TN / (TN + FP).
- FPR (False Positive Rate) : FP / (FP + TN).
- FNR (False Negative Rate) : FN / (TP + FN).
- PWC (Percentage of Wrong Classifications) : 100 * (FN + FP) / (TP + FN + FP + TN).

F-Measure

- : (2 * Precision * Recall) / (Precision + Recall).
- Precision : TP / (TP + FP).

While each metric gives a different insight into the results, the F-Measure is the most commonly used one. Therefore, we mainly use F-Measure to evaluate the DFFnetSeg.

The LASIESTA dataset is composed of 17 real indoor and 22 outdoor sequences organized in 12 categories, including simple sequences (SI), camouflage (CA), occlusions (OC), illumination changes (IL), modified background (MB), bootstrap (BS), moving camera (MC), simulated motion (SM), cloudy conditions (CL), rainy conditions (RA), snowy conditions (SN), and sunny conditions (SU). Same as CDnet2014, we also exclude MC and SM categories, because those camera motion patterns are not in the scope of the DFFnetSeg.

TABLE 2. Scenes and frame indices used in testing on CDnet2014 dataset.

Category	Sequence	Frame indices	Category	Sequence	Frame indices
Sh	backdoor	1490-1989	DB	canoe	800-1189
	peopleInShade	574-1073		overpass	2306-2805
TH	park	250-600	IOM	parking	1291-1790
	diningRoom	1241-1740		winterDriveWay	1722-2221
BW	snowFall	2701-3200	NV	streetCornerAtNight	2396-2895
CJ	traffic	1050-1549		tramStation	1237-1736
LF	turnpike_0_5fps	800-1149	PTZ	twoPositionPTZCam	820-1045
Ba	pedestrians	306-805			1100-1380

B. TRAINING AND TESTING SET

Many state-of-the-art deep learning fg/bg classification algorithms generate the training set by separate each sequence by half, 80% or selected number, with the rest sequence frames as the testing set, in which the same background scenes and foreground objects have chance to coexist in both training and testing set. However, in our experiment, we generate the training and testing set based on the principle that the background scenes and foreground objects in the testing set have no overlap with the corresponding in training set. In detail, we use 1 to 2 sequences in each category, including *backdoor*, *canoe*, *dinningRoom*, *overpass*, *park*, *parking*, *pedestrians*, *peopleInShade*, *snowFall*, *streetCornerAtNight*, *traffic*, *tramStation*, *turnpike_0_5fps*, *twoPositionPTZCam*, and *winterDriveway* from CDnet2014 and *I_BS_01*, *I_CA_01*, *I_CA_02*, *I_IL_01*, *I_MB_01*, *I_MB_02*, *I_OC_01*, *I_SI_01*, *I_SI_02*, *O_CL_01*, *O_CL_02*, *O_RA_02*, *O_SN_01*, and *O_SU_01* from LASIESTA as testing sequences and the remainder as training ones. For training, we randomly choose around 1000 frames from each training sequence (when the available frames with ground truth are less than 1000, we use all the frames). For testing, we randomly select one clip from each sequence and based on these clips, we construct the testing set by two parts. The first part of the testing set consists of single test clips as sequences, which is mainly to test the generality of algorithms. In the second part, we simulate the surveillance video captured by the camera whose position or location changes 1, 2 and 3 times by randomly concatenating 2, 3 and 4 test clips together, which is mainly to test the robustness of algorithms when scene changes occur in sequences. In terms of the implementation details, we consider 30 test clips in all which is not exactly divisible by 4, so 2 remainders are abandoned when concatenating 4 clips.

The details of selected clips from CDnet2014 dataset are shown in Table 2. 500 continue frames including foreground are randomly chosen from each sequence, when the sequence is longer than 500 frames. Otherwise, the whole sequence is chosen as the testing clip. Under most circumstance, 500 frames are enough for most background initialization method to estimate the background model, and are also short enough to avoid long term static scene, which is proper to evaluate the robustness of algorithms to the scene change.

The details of selected clips from LASIESTA are shown in Table 3. The frame with big enough foreground object is chosen as the first frame for each sequence, which brings

TABLE 3. Scenes and frame indices used in testing on LASIESTA dataset.

Category	Sequence	Frame indices	Category	Sequence	Frame indices
BS	I_BS_01	1-275	CA	I_CA_01	105-350
IL	I_IL_01	110-300		I_CA_02	193-525
MB	I_MB_01	110-450	SI	I_SI_01	88-300
	I_MB_02	100-350		I_SI_02	97-300
OC	I_OC_01	110-250	SN	O_SN_01	336-500
CL	O_CL_01	150-225	RA	O_RA_02	186-375
	O_CL_02	118-425	SU	O_SU_01	122-250

a general challenge *bootstrap* to background subtraction. It is because different from the single-scene video which generally has tons of history frames to obtain a relatively clean background (without foreground), multi-scene videos are difficult to obtain informative history frames to extract background. Therefore, multi-scene video foreground object detection method should have the ability to deal with the situation with *bootstrap* challenge.

C. RESULTS

To evaluate the effectiveness of DFFnetSeg method, we compare it with the following state-of-the-art algorithms:

- SubSENSE [7], PAWCS [14] and SWCD [8], the top traditional background subtraction methods on CDnet2014 with source code open to the public.
- FgSegNet [11], the top foreground segmentation method on CDnet2014 with source code open to the public.
- BScGAN [18], the deep background subtraction algorithm with conditional generative adversarial networks which reports a top result on CDnet2014 dataset in the original paper.

on two datasets mentioned above. To ensure a fair comparison between the supervised models, we use the same training data as our model to train the models and the hyper-parameters are same as the ones described in the source code and paper. The pre-trained model is also used to initialize the model parameters if the initialization is mentioned in original papers.

In order to assess the DFFnetSeg for a large set of parameters in region-based motion map stage mentioned in Section III-B, an estimation of the foreground mask has been generated for all the single test clips using each combination of $\theta \in \{20, 50, 80, 110\}$, $N \in \{8, 16, 32, 64\}$, $\beta \in \{5, 10, 20, 40\}$. The three parameters of DFFnetSeg method (θ, N, β) provide enough flexibility to boost the foreground mask for various input video sequences. The raw result of the fusion network mentioned in SectionIII-A is with average f-value 0.8448. When we use the pixel-level motion map ($N = 1$, do not consider β), the best result with the parameters mentioned above is worse than the raw one with average f-value 0.84 ($\theta = 20$). When $(\theta, N, \beta) = (20, 16, 20)$, the region-based motion map achieves the largest boost effect with f-value 0.884. In detail, when the considered parameters are in the range shown in Table 4, the region-based motion map boosts the foreground mask in different degrees. Otherwise, it wrecks the final prediction. The table shows that $\theta = 110$ is too big to catch the motion pixels; the greater the θ, the greater the N, and the less the β is demanded to extract the good enough motion map.

TABLE 4. Average F-measure on the single test clips with different parameters range for the motion map.

Parameters Range			Best F-Measure	Worst F-Measure
$\theta = 20$	$N > 8$	$\beta \in U$	0.884	0.863
$\theta = 50$	$N > 8$	$\beta \leq 10$	0.881	0.855
	$N > 16$	$\beta > 10$	0.880	0.867
$\theta = 80$	$N > 16$	$\beta = 5$	0.874	0.846
	$N > 64$	$\beta > 5$	0.871	0.855

TABLE 5. Average F-measure on all the test sequences with different γ for the reference frame updating.

γ	F-Measure
no updating	0.7296
0.4	0.8522
0.5	0.8529
0.6	0.8527
0.7	**0.8531**
0.8	0.8494
0.9	0.8450

To choose the best parameter for scene change detector we test all the testing sequences including the single clips and concatenated sequences using $\gamma \in \{0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, as shown in Table 5. As γ is the threshold for the level of similarity of frames, when γ is high, the reference frame updating is more sensitive to the scene difference and vice versa. As shown in Table 5, when $\gamma = 0.7$, the updating strategy promotes the performance to the best, whereas when γ becomes higher, the result becomes worse because the scene change detector is so sensitive that discriminates the scene change by mistake when the large foreground is moving.

D. IMPLEMENTATION DETAILS

In our experiments, only the fusion network part need to be trained in a supervised way by labelled change detection data, as described in Section III-A. The PSPNet is pre-trained on ADE20K as in [26] and no parameter inside the PSPNet is fine-tuned. As the purpose of the fusion network is to generate the foreground object mask based on the frames difference and current frame contents, the reference frame is fixed to be $f^{(1)}$ during the training and the parts proposed in Section III-B and Section III-C are only for testing. Before the frames are fed to PSPNet, the size of frames are standard to 473×473 and subtracted by the mean as preprocessing. The initial learning rate is 10^{-4} and we use Adam for optimisation.

Our experiment is implemented based on TensorFlow framework on a single NVIDIA GeForce GTX 1080ti GPU. The training process was completed in about 10 hours with 5 epochs.

E. COMPARISON

We compare the performance among methods on the testing data with and without halfway scene changes separately.

TABLE 6. Evaluation values of models on single clips part of the test dataset.

Method	Recall	Specificity	FPR	FNR	PWC	Precision	FM
SubSENSE [7]	0.6578	0.9929	0.0071	0.3422	2.8384	0.8209	0.7026
PAWCS [14]	0.6029	0.9973	**0.0027**	0.3971	2.7259	**0.8936**	0.6779
SWCD [8]	0.6325	0.9845	0.0155	0.3675	3.4388	0.7568	0.6447
FgSegNet [11]	0.7663	0.9233	0.0767	0.2337	8.8115	0.4473	0.4883
BScGAN [18]	0.8558	0.9802	0.0198	0.1442	2.8641	0.6329	0.6979
DFFnetSeg	**0.9312**	**0.9935**	0.0065	**0.0688**	**1.1739**	0.8602	**0.8845**

TABLE 7. Evaluation values of models on multi-scene sequences part of the test dataset.

Method	Recall	Specificity	FPR	FNR	PWC	Precision	FM
SubSENSE [7]	0.6368	0.9394	0.0606	0.3632	7.4710	0.3694	0.4407
PAWCS [14]	0.6396	0.8560	0.1440	0.3604	15.4037	0.2072	0.2822
SWCD [8]	0.7131	0.9487	0.0513	0.2869	6.2381	0.5112	0.5596
FgSegNet [11]	0.8032	0.9221	0.0779	0.1968	8.2251	0.3918	0.5004
BScGAN [18]	0.8039	0.9644	0.0356	0.1961	4.3326	0.4983	0.5955
DFFnetSeg	**0.9263**	**0.9921**	**0.0079**	**0.0737**	**1.0740**	**0.8509**	**0.8826**

The performance on the sequences without halfway scene changes shows the universality of methods under different challenges, while the decay level of the performance when the scene change is included in testing videos shows the robustness of methods to the halfway scene change challenge. As shown in Table 6, SubSENSE, PAWCS, SWCD and BScGAN have similar performance, whereas the DFFnet dramatically outperforms these methods with a much higher F-Measure value. The performance of FgSegNet is extremely bad because it is originally proposed as a scene-specific method to predict foreground segmentation based on a single frame, which makes it easy to lack universality. In terms of the multi-scene testing, as shown in Table 7, the performance of SubSENSE, PAWCS, SWCD and BScGAN decreases significantly, especially PAWCS, but the corresponding of DFFnetSeg keeps steady with only 0.2% F-Measure drop. By contrast, the performance of FgSegNet does not change because it is based on a single frame which has no relationship with the scene change. The slight increase of the F-Measure is actually caused by the missed video clips as mentioned in Section IV-B. Except the DFFnetSeg, the SWCD and BScGAN are more robust with only 8.5% and 10.2% F-Measure drop respectively and the latter BScGAN gets the relative better F-Measure when the scene change happens, which benefits from the background modelling method it chooses and the GAN architecture.

More details are shown in the sampled visualization Fig 4. We choose the sequences from different typical challenges for comparison (category details in Table 2 and Table 3). As we can see from the figure, the DFFnetSeg outperforms others almost in all samples, from the perspective of both accurate classification and clear edges. In addition, the DFFnetSeg still performs well even when the scene changes halfway in the sequence. However, the testing data seem quite challenging for other methods. As the FgSegNet only consider a single frame for foreground segmentation, it highly depends on object detection rather than motion detection. Therefore, a great number of false positives caused by wrong object classification exist in its output masks but the output masks are totally not influenced by the scene change. The rest 4 methods as background subtraction methods all need to construct the background model, so the scene change makes a great impact on their performance. Therefore, we discuss the result of these 4 methods before and after the scene changes separately.

Firstly, we analyze the performance before the scene change happens. Notice, the performance of the existing methods is different from the one shown on the benchmark website because, in their experiments, the beginning frame of the same video is different from the one in our experiment. In our experiment, most beginning frames contain foreground objects inside which is a more common situation in nature and more challenge than the first frames without foreground objects. For example, the 2nd and 3rd row in the figure are frames from the same sequence, but the performance of SubSENSE, PAWCS and SWCD shown in the 3rd row is much better than the corresponding in the 2nd row, which indicates that those methods take time to construct the fine and stable background models. It aligns well with the mechanism of most background subtraction methods which generally need different length of video clips to construct their stable enough background models. Except FgSegNet, the rest state-of-the-art methods are competitive with each other. For example, BScGAN performs better in *winterDriveway* whereas PAWCS performs better in *overpass*. Although the *peopleInShade* is in Sh category, the bad results are actually caused by the stopped foreground which stopped since the 34th frame in the test clip (the current frame is the 137th in that clip).

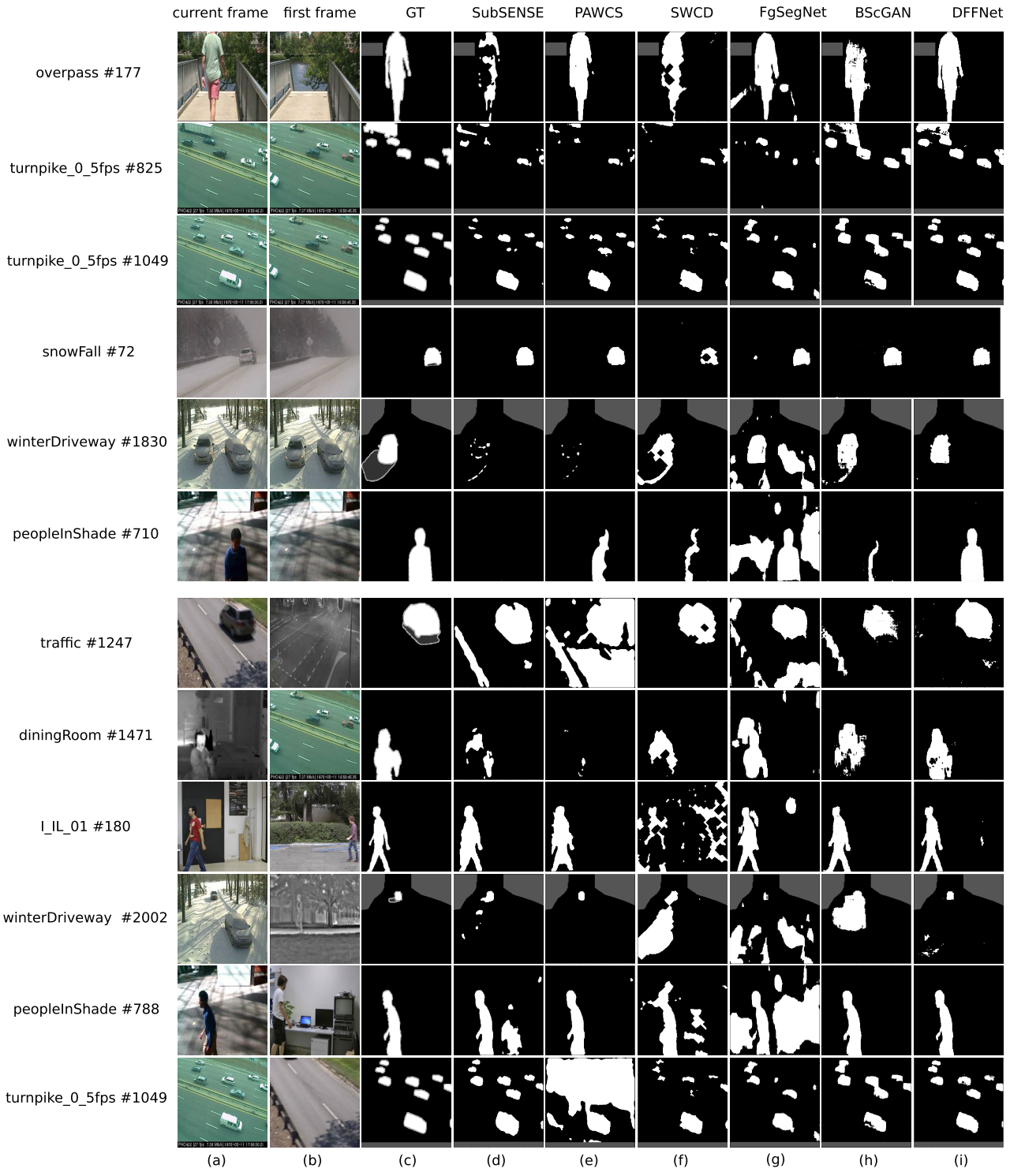


FIGURE 4. Qualitative results comparison on the 6 algorithms. The first 6 rows show results before the scene changes, while the last 6 rows show results after the scene changes. The number behind # denotes the indices of the frame in original sequence from the dataset, so that the corresponding indices of the frames in our clips can be obtained by subtracting the begin indices. For example, the corresponding indice of the first row is 88.

In conclusion, for sequences without scene changes, the qualitative result aligns well with the quantitative result that SubSENSE, PAWCS, SWCD, and BScGAN have similar performance.

Secondly, we analyze the performance after the scene change happens. It is obviously PAWCS is not as good at dealing with the scene change as others, because of its large scale false positives which are caused by the bad background

TABLE 8. Evaluation values of the model if a component is removed.

	Single Clips			Multi-scene Sequences		
Method	F-Measure	Recall	Precision	F-Measure	Recall	Precision
Fusion Network	0.8448	**0.9417**	0.7943	0.6912	0.8931	0.5779
+Motion Map	0.8840	0.9315	0.8596	0.6973	0.8886	0.5887
+Reference Updating	0.8453	0.9415	0.7951	0.8557	**0.9368**	0.7960
All	**0.8845**	0.9312	**0.8602**	**0.8826**	0.9263	**0.8509**

model. For the other methods, they still take time to reconstruct the background model, and the performance is acceptable after reconstructing a proper background model. From the perspective of visual evaluation, BScGAN performs better than other methods, which benefits from both the background modelling method it utilizes and its network robustness to background noise.

Actually, the qualitative evaluation is limited by sampling frames from sequences because the performance quality of methods changes a lot along the time. It takes different frame numbers for different methods to reconstruct a proper background model which can also be understood as the *speed* of background model reconstruction, which can not be shown by simply sampling. Nevertheless, quantitative evaluation can properly evaluate this performance feature which changes along the time sequence. Therefore, the high quantitative evaluation result comes not only from the fine classification but also from the fast scene change response mechanism.

In conclusion, the DFFnetSeg outperforms the state-of-the-art methods by a great gap in both quantitative evaluation and qualitative evaluation.

F. ABLATION STUDIES

In this subsection, we justify the decisions we made in the DFFnetSeg by conducting a series of ablation tests. In particular, we evaluate the performance of the DFFnetSeg by testing the effect of removing individual components on foreground mask generation tasks. The results in Table 8 are obtained by parameters $(\theta, N, \beta, \gamma) = (20, 16, 20, 0.7)$. The main evaluation value F-Measure shows the increasing trend when the proposed components are added. Table 8 shows that for the single clips the motion map has a great positive impact on the performance with F-Measure, increasing from 0.8449 to 0.884, because it can reduce the false positives caused by semantic noise but it demands the reference frame and the previous frame are from the same scene as the current frame. Therefore, without the reference frame updating, the motion map almost does not contribute to the foreground mask when the scene change happens in the sequence. On the other hand, the reference frame updating makes great contribution to the multi-scene sequences (F-Measure from 0.6912 to 0.8557) but rarely contributes to the single clip results. It is because the reference frame updating strategy is designed to deal with the multi-scene sequences. Besides, when no scene change happens in the sequence, our fusion network performs also good even with the noisy frame (frame with foreground object) as the reference frame. Notice, for multi-scene sequence, only adding the motion map does not boost our performance but when the motion map combines with the updating strategy, our performance is improved further.

V. CONCLUSION

We propose a robust foreground segmentation approach based on deep features fusion network by using features extracted from a semantic segmentation network PSPNet in a comparison and fusion architecture. By contrast to other semantic-based background subtraction methods, our fusion network learns to combine the semantic information of the current frame with the soft motion map extracted from the current frame, the previous frame, and the reference frame. By contrast to other deep learning methods, DFFnetSeg generates high-quality foreground masks on not only the unseen videos but also the multi-scene videos.

As the DFFnetSeg method is designed for position or location changed surveillance camera videos and takes the advantage of semantic information to get high-quality foreground mask, in future, it has potential to be extended to the continue position changing surveillance camera videos.

ACKNOWLEDGMENT

The authors would like to thank the benchmark website (www.changedetection.net) and the dataset LASIESTA.

REFERENCES

[1] 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.

[2] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, "Progressive search space reduction for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[3] T. Chen, W. Yin, X. S. Zhou, D. Comaniciu, and T. S. Huang, "Illumination normalization for face recognition and uneven background correction using total variation based image models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 532–539.

[4] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–8.

[5] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 38–43.

[6] K. Lim, W.-D. Jang, and C.-S. Kim, "Background subtraction using encoder-decoder structured convolutional neural network," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2017, pp. 1–6.

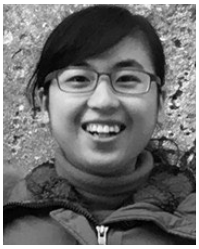
[7] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[8] S. Isik, K. Özkan, S. Günal, and Ö. N. Gerek, "SWCD: A sliding window and self-regulated learning-based background updating method for change detection in videos," *Proc. SPIE*, vol. 27, no. 2, Mar. 2018, Art. no. 023002.

[9] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split Gaussian models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 414–418.

[10] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognit.*, vol. 76, pp. 635–649, Apr. 2018.

[11] L. A. Lim and H. Y. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognit. Lett.*, vol. 112, pp. 256–262, Sep. 2018.
[12] 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.
[13] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jun./Jul. 2015, pp. 1–6.
[14] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Jan. 2015, pp. 990–997.
[15] 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.
[16] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2105–2115, Sep. 2017.
[17] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, May 2016, pp. 1–4.
[18] M. C. Bakkay, H. A. Rashwan, H. Salmane, L. Khoudour, D. Puigtt, and Y. Ruichek, "BSCGAN: Deep background subtraction with conditional generative adversarial networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 4018–4022.
[19] W. Zheng, K. Wang, and F. Wang, "Background subtraction algorithm based on Bayesian generative adversarial networks," *Acta Autom. Sinica*, vol. 44, no. 5, pp. 878–890, 2018.
[20] W. Zheng, K. Wang, and F.-Y. Wang, "A novel background subtraction algorithm based on parallel vision and Bayesian GANs," *Neurocomputing*, to be published.
[21] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixelwise deep sequence learning for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2567–2579, Sep. 2019.
[22] M. Jaderberg, K. Simonyan, and A. Zisserman, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
[23] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.
[24] 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.
[25] M. Braham, S. Piérard, and M. Van Droogenbroeck, "Semantic background subtraction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4552–4556.
[26] 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.
[27] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[28] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5122–5130.
[29] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.
[30] C. Cuevas, E. M. Yáñez, 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.



YE TAO received the B.S. degree in measurement and control technology and instruments from the East China University of Science and Technology, China, in 2014, where she is currently pursuing the Ph.D. degree.

Her research interests include background subtraction, foreground segmentation, deep learning, and background modeling.



ZHIHAO LING received the B.S. degree in automation and the Ph.D. degree in control science and engineering from the East China University of Science and Technology, China, in 1982 and 2005, respectively.

He is currently a Professor with the East China University of Science and Technology. His research interests include the Internet of Things, wireless sensor networks, embedded application systems, detection technology, and instrument intelligence.



IOANNIS PATRAS received the B.S. and M.S. degrees from the University of Crete, Heraklion, Greece, in 1994 and 1996, respectively, and the Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 2001, all in computer science.

He is currently a Professor of computer vision and human sensing with the School of EECS, Queen Mary University of London, U.K. His research interests include human sensing, affective computing, computer vision, machine learning, and deep learning.