IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 26, 2018, accepted March 4, 2018, date of publication March 19, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817129

Background Subtraction Using Multiscale Fully Convolutional Network

DONGDONG ZENG[1,2,3] AND MING ZHU[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]Key Laboratory of Airborne Optical Imaging and Measurement, Chinese Academy of Sciences, Changchun 130033, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Dongdong Zeng (dongdong.zeng@gris.tu-darmstadt.de)

This work was supported by the National Science Foundation of China under Grant 61401425.

ABSTRACT Background modeling and subtraction based on change detection are the first step in many high-level computer vision applications. Many background subtraction methods have been proposed in the recent past and their efforts mainly focus on two aspects: more advanced background models and more complex feature representations. Recently, hierarchical features learned from deep convolutional neural networks have been shown to be effective for many computer vision tasks, such as classification and recognition. However, few researchers try to learn the deep features to address the background subtraction problem. Therefore, in this paper, we propose a novel multiscale fully convolutional network (MFCN) architecture which takes advantage of different layer features for background subtraction. We show that the foreground detection accuracy can be greatly improved by using the deep features learned from the MFCN and instead of building highly complex background models, and the complexity of the background subtraction process can be easily solved during the subtraction operation itself. Experimental results on CDnet 2014 data set and SBM-RGBD data set show that the proposed MFCN-based method achieves state-of-the-art performance while operating at real time.

INDEX TERMS Background subtraction, convolutional neural network, multiscale fully convolutional network, video surveillance.

I. INTRODUCTION

Background subtraction based on change detection is the first step in many high-level computer vision systems. The output of the background subtraction is usually an input to a post higher level process, such as traffic monitoring, object tracking, and action recognition. Therefore, the accuracy of the detection result has a huge effect on these subsequent higher level tasks. Needless to say, the quality of many computer vision applications directly depends on the quality of the background subtraction method used.

Generally speaking, a complete background subtraction process has four components (See Fig. 1): (1) model initialization, which regards the initialization process; (2) model representation, which describes what kind of model to be used to represent the background model; (3) model maintenance, which concerns the update mechanism used for adapting the model to the changes; (4) foreground detection, which consists of comparing the current frame with the background image and classifying the pixels as foreground or background. In the past few decades, a multitude of background subtraction methods have been proposed and have achieved promising progress [1]–[3]. However, it is still regarded as a challenging problem due to the complexity of environment such as dynamic backgrounds, illumination changes, shadows, camera jitter, camouflage, and so on. It is not straightforward to handle all these challenges in a single framework.

Recently, convolutional neural networks (CNNs) have drawn a lot of attention in the computer vision community. Hierarchical features learned from deep convolutional neural networks have been shown to be effective for many computer vision tasks such as classification [4], [5], recognition [6], [7], semantic segmentation [8], [9], saliency detection [10], [11], and so on. Despite its popularity, only a few researchers are attempting to employ the CNNs for background subtraction. To the best of our knowledge, [12] is the first attempt to apply CNN to solve the background subtraction problem on a single scene and an improved CNN-based method with a novel background image generation was proposed in [13]. Another multi-scale and cascade convolutional neural network based method was proposed in [14] and

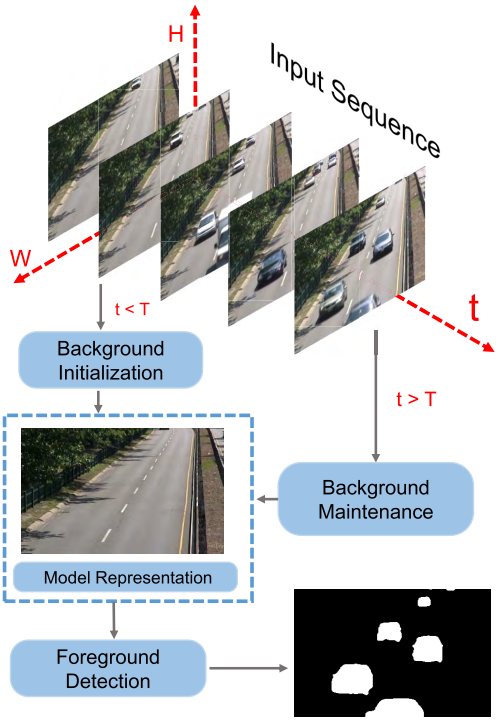


FIGURE 1. Block diagram of the background subtraction process.

achieved state-of-the-art results on the CDnet 2014 dataset at present.

However, current CNN-based approaches have several drawbacks. First, it is quite slow for these patch-wise based methods because the network must run each patch separately, so it is difficult to achieve real-time performance, and it also result in a lot of redundancy due to the overlapping patches. Second, using neural networks to classify the background and foreground, only the outputs of the last layer are considered following the recent object recognition tasks. As we know, for high-level visual recognition problems, it is more effective to use features from the last layer as they are more closely related to category-level semantic information. However, for foreground detection, there is a trade-off between the segmentation accuracy and the use of the semantic information. Third, training a robust classifier requires a large number of training samples, while this is not available in background subtraction area. Current CNN-based methods use highly redundant scene-specific data for training will lead to overfitting problems.

Inspired by the above observations, in this paper we propose a novel multiscale fully convolutional network (MFCN) architecture for background subtraction. Our method draws on the recent success of transfer learning and fully convolutional network (FCN) for semantic segmentation. Transfer learning has recently been applied to many areas [15]–[17]. Previously trained network weights are used to initialization and then the weights are fine-tuned on a new dataset. Compared with training a network from random initialization, this method can sometimes achieve better accuracy. The FCN architecture was first proposed in [8] for image segmentation. By transforming fully connected layers into convolutional layers, the new architecture can be trained end-to-end. Compared with patch-wise methods, the FCN-based models can capture more local and global context information, which yields more accurate and detailed segmentation results. And recent research shows that deep features obtained from different convolutional layers can improve the results for different image tasks [5], [16]. The lower layers contain low-level semantic information but retain higher spatial resolution, while the deep layers capture more high-level semantic information but with less spatial details. Taking features from different convolutional layers into account can get more precise localization and achieve high-level semantics at the same time. Thus in this paper, we re-architect and fine-tune the VGG-16 [5] network and use the fully convolutional network. Multiscale convolution and deconvolution operations are used to make the output of the network has the same size with the input while capturing the local and global context as well as features at various resolutions to make a more accurate foreground segmentation result.

Our contributions lie in two aspects: first, we propose a novel multiscale fully convolutional network (MFCN) architecture which takes advantage of different layer features for background subtraction. Extensive experiments are performed on the CDnet 2014 and SBM-RGBD dataset. The results show that the proposed approach is superior to the existing state-of-the-art methods and also shows real-time performance. Second, we show that in contrast to traditional background subtraction methods which contain complex background modeling and updating processes, our method can simplify these steps into a simple network classification process.

The rest of this paper is organized as follows. Section II gives a brief introduction of related works. Section III describes the framework of the proposed MFCN-based background subtraction method. Section IV shows the experimental results carried out on the CDnet 2014 and SBM-RGBD dataset compared with other state-of-the-art methods. Final conclusions are given in Section V.

II. RELATED WORKS

There are various kinds of background subtraction algorithms, and it is difficult to review all prior works here. Readers are recommended to refer to [1] and [2] for a thorough review of background subtraction. In this section, we focus on several traditional background subtraction algorithms first and then discuss more recently CNN-based background subtraction algorithms.

A. TRADITIONAL BACKGROUND SUBTRACTION

Traditional background subtraction methods mainly manifest in two aspects. The first one is to construct more advanced background models. For example, a very classic and popular background subtraction method GMM was proposed in [18], which models each pixel with a mixture of

Gaussian. As further development, more flexible and adaptive variations were proposed in [19] and [20] to improve the model update speed and the model stability. In [21], Elgammal et al. proposed a nonparametric approach based on kernel density estimation (KDE), which directly estimates the pixel probability distribution function from the data without any prior assumptions. The algorithm presented in [22] and [23] named SOBS implements a background subtraction approach based on self-organizing neural networks and achieves good results in various situations. ViBe [24] presented a non-deterministic background subtraction method, a stochastic update strategy is used to integrate new scene information into the background model. An improvement of ViBe called SuBSENSE was proposed in [25] which combines the color and local binary similarity pattern features to improve the spatial awareness of change detection.

The second one is to employ a more powerful feature representation such as color features, edge features, motion features, and texture features. Color features are the most commonly used to characterize pixel representations, however, they have some limitations in the presence of challenges such as camouflage, shadows, and illumination changes. To deal with these challenges, other features like edge [26], motion [27] and texture [28], [29] features are proposed. The edge features have the merit of dealing with local illumination change and the ghost problems. The motion features are usually calculated via optical flow, so it is computationally expensive. The local binary pattern (LBP) feature [28] is the first texture feature proposed for background subtraction. An improved version called scale invariant local ternary pattern (SILTP) was proposed in [30], which exceed LBP in the computational efficiency and tolerance to noises. In [31], an extended scale invariant local binary pattern called ESILBP was proposed and shows considerable robustness for image noise and illumination variations. Recently, local bianry similarity pattern (LBSP) was proposed in [25] and [29], which took the spatiotemporal information into consideration to enhance the feature discriminant performance.

B. CNN-BASED BACKGROUND SUBTRACTION

The first CNN-based background subtraction approach was proposed in [12]. The key idea of their method is: firstly, a gray scale background image is extracted from several initialization frames with the temporal median operation. Then, for each pixel, two small patches with a size of 27×27 centered on the pixel are extracted from the input frame and the background image. Finally, feeding the patches through the trained network to compute the foreground probability for that pixel. However, generating the background image through the temporal median is not always feasible. It is appropriate only when each background pixel is visible for more than 50% of the time. Babaee *et al.* [13] propose a novel approach for background image generation by combining the segmentation mask from [25] and [32] to alleviate this issue. Another CNN-based background subtraction was proposed in [14] with a multi-scale and cascade convolutional neural network architecture. The input frame is first downsampled into two different scales with 0.75 and 0.5. Then the original input frame and the scaled frames are fed to a basic CNN network which contains 4 convolutional layers and 2 fully connected layers. Finally, the different scale output maps are resized back to the input size and the final foreground segmentation map is obtained with an average pooling operation on these upscaled maps. This approach achieved state-of-the-art results on the CDnet 2014 dataset at present.

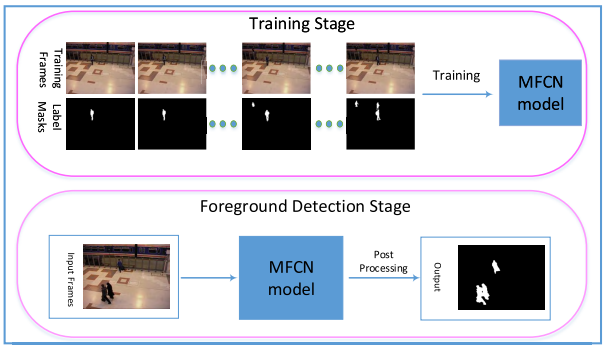


FIGURE 2. Process of the proposed MFCN-based background subtraction algorithm.

III. MFCN-BASED BACKGROUND SUBTRACTION

In this section, we will give a detailed description of the framework for the multiscale fully convolutional network (MFCN) based background subtraction method. Fig. 2 shows the pipeline of our method, which contains two stages. The training stage and the foreground detection stage. During the training stage, a few input frames with their corresponding foreground/background label masks are used to train the model. Once the training process is over, the MFCN model is used for foreground segmentation across all sequences.

A. TRAINING DATA PREPARATION

In order to train our network, we have two ways to get the training data. Depending on the way of the training data is generated, the final background subtraction algorithms can be viewed as supervised and unsupervised.

1) **Supervised:** The training frames and their corresponding label masks are taken from the Change Detection challenge benchmark dataset, whose ground truths are constructed by the human expert. For example, in the CDnet 2014 dataset [33], for each sequence, we consider the first 3000 frames (with the ground truths available) as the initialization process (this is about 2 minutes length of video with the frame rate of 25fps). Then, a subset of 200 frames is randomly and manually selected from these frames. Finally, the selected frames with their corresponding ground truths are used to train the model. If the sequence frame number is less than 3000, the training frames are selected across the whole frames. This strategy has a disadvantage that it

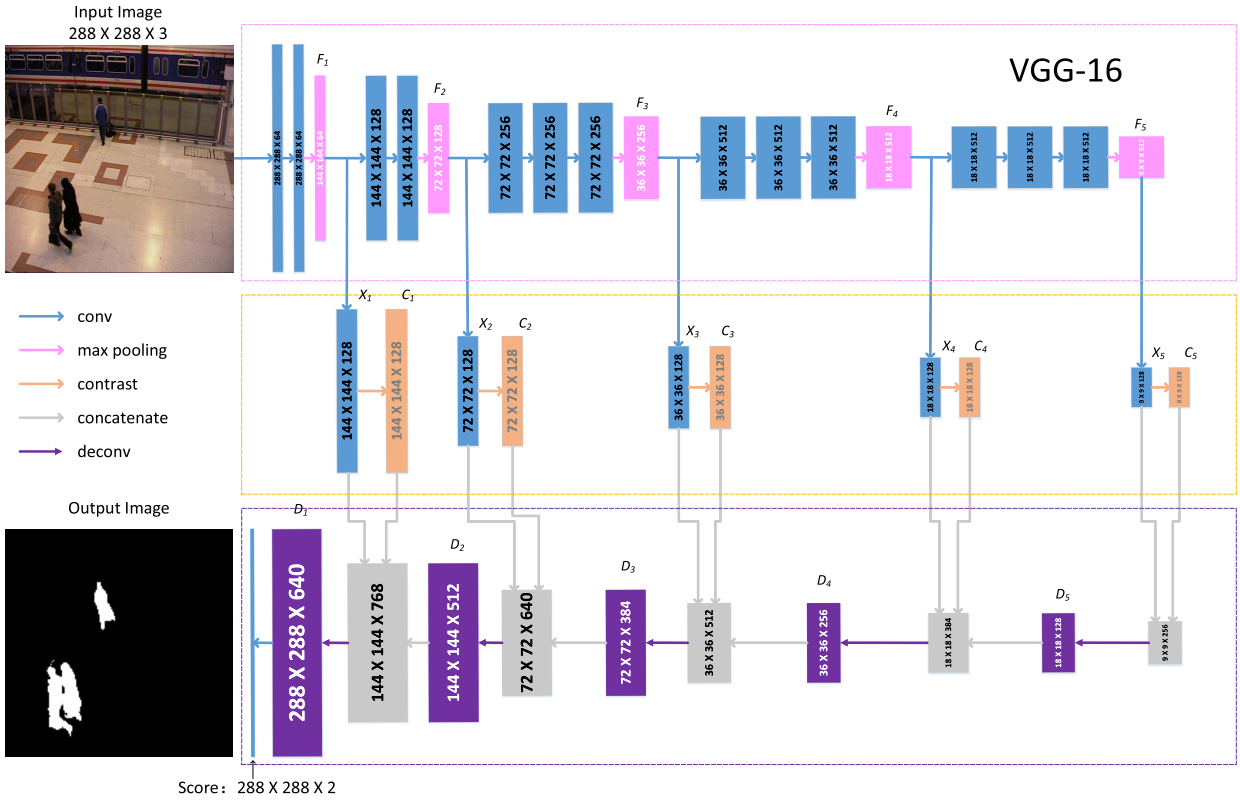


FIGURE 3. Architecture of the proposed MFCN for background subtraction. This architecture is based on the VGG [5] network, which is separated into five stages by max pooling operations. In order to effectively utilize multiscale features from different stages, a set of convolution and deconvolution operations with the stepwise upsampling strategy are used to aggregate multiscale features, making a feature representation that contains more category-level information and fine-grain details.

requires human intervention, but it can greatly improve the detection accuracy.

2) **Unsupervised:** The training frames and their corresponding label masks are taken from the results generated by other existing background subtraction algorithms. For example, we can use the results generated by PAWCS [34], which is currently average ranked the first on the CDnet 2014 dataset, as the training data. This strategy has the advantage that it is unsupervised, without human intervention. However, the disadvantage is that the final classification accuracy is determined by the performance of the chosed algorithm.

In this paper, we mainly focus on training the network in a supervised manner. After all the training frames and the label masks have been collected, pre-processing is performed on these data. As we will show in section III-B, our network is based on the VGG-16 [5] network architecture and the inputs are with a size of 288 × 288 × 3, so we have to pre-resize all training frames to the fixed size 288 × 288 × 3 (for gray images, we just set the R, G and B channels equal to the gray intensities) and then a mean subtraction is operated on each pixel. For the label masks, since we treat the background and foreground segmentation as a binary classification problem, thus the corresponding training label masks will have two channels (2 classes) with a size of 288 × 288 × 2. The label value is given by:

$$y_p = \begin{cases} 1, & \text{if } class(p) = foreground; \\ 0, & \text{otherwise.}, \end{cases} \tag{1}$$

where p denotes the pixels in the label masks.

B. NETWORK ARCHITECTURE

The architecture of the proposed MFCN is shown in Fig. 3. A fully convolutional network architecture with multiscale convolution and deconvolution operations. Unlike the previous works [12], [13], our method has no need to extract the background images. The input of our network is the RGB frame from different sequences, the output is a probability map (one channel, the size of which is the same as the input). Here we organize our network into three parts: the pink dashed box (the first row), the orange dashed box (the second row) and the purple dashed box (the third row).

The pink dashed box contains the VGG-16 [5] network. As in our experiment, only a small number of training data is available, therefore we fine-tune our model on the VGG-16 network. Compared with training a new network from random initialization, this method can result in better accuracy. We can split the VGG-16 network into 5 stages with each

containing some convolution and max pooling operations. The sizes of the corresponding output layers are shown in Fig. 3. We can see that the lower layers have higher spatial resolutions but perceive only low-level local features, while the deeper layers can perceive more high-level global features but are with lower resolutions. Be different with the original VGG-16 network, we have to make some modifications so that it is suitable for our tasks: firstly, we cut the last stage of the VGG-16 network, including all the fully connected layers. The fully connected layers contain more high-level semantic information but with less spatial details, which is unsuitable for the background subtraction task. And the fully connected layers are computationally expensive, trimming these layers can significantly reduce the complexity of our model and improve the detection speed. Secondly, as we mentioned earlier, taking features from different convolutional layers into account can not only get more precise localization and but also achieve high-level semantics at the same time. Then, in order to aggregate these multiscale features from different feature layers, 3×3 convolution kernels are operated on the max pooling output layer in each stage, respectively F_1, F_2, F_3, F_4 and F_5 in Fig. 3. The output convolution feature maps (X_1, X_2, X_3, X_4 and X_5) keep the same spatial resolution with the upper layers and all of them have 128 channels.

The orange dashed box contains the multiscale features (X_1, X_2, X_3, X_4 and X_5) extracted from different stages of VGG-16 and the contrast features (C_1, C_2, C_3, C_4 and C_5). The contrast feature layers are used to extract the outstanding difference between foreground object and its local neighborhood region. Since in this paper, we treat the background and foreground segmentation as a binary classification problem. From the output binary masks we can see that there is a great contrast between the foreground objects and their backgrounds, and in most cases the foreground objects are uniform connected areas, which means that in the input frames, there should also have a big difference of the features between the foreground and the background. So in order to extract this kind of contrast information, we add a contrast layer after the feature layer X_i. The contrast feature C_i is calculated as follows:

$$C_i = X_i - avgPool(X_i), \tag{2}$$

where $avgPool(X_i)$ is the average pooling operation on the feature X_i with a kernel size of 3×3. Then the difference between X_i and its local average result $avgPool(X_i)$ represents the contrast information.

After getting the multiscale features from different layers, a set of deconvolution operations is used to upsample these feature maps to make the final output probability map has the same size as the input, as shown in the purple dashed box of Fig. 3. However, instead of upsampling the feature maps with a fixed ratio in [8], e.g: 8x, 16x and 32x. We adopt a stepwise upsampling strategy to produce more refined feature maps. Firstly, the feature map X_5 from the fifth stage of the VGG-16 network is concatenated with its contrast feature C_5 on the last dimension. The concatenated feature is upsampling by a factor of 2 with the deconvolution operation to get the new deconvolution layer D_5. Then, the feature D_5 is concatenated with the feature map X_4 and its contrast feature C_4 from the fourth stage of VGG-16. Upsampling the concatenated feature as before and get the next stage deconvolution layer D_4. After five stages of deconvolution operations, the features maps with different scales are integrated and upsampled to the input size. These operations can be expressed as follows:

$$D_{i-1} = Deconv(Concat(X_i, C_i, D_i)). \tag{3}$$

The *Concat* is the concatenation operation of the feature maps on the last dimension and the *Deconv* is the deconvolution operation with a kernel size of 3×3 and stride is 2. In the end, the last deconvolution feature layer D_1 has a size of $288 \times 288 \times 640$. Then, convolution kernel with a size of 1×1 is operated on D_1 and producing a score layer which contains two channels. Finally, a softmax operation is performed on the score layer to get the final foreground probability map.

For the loss function, due to the distribution of foreground/background pixels is heavily biased, we use the class-balancing cross-entropy loss, which was firstly proposed in [35] for contour detection tasks. Let's denote the training data as $S = \{(I_n, Y_n), n = 1, \ldots, N\}$,, where I_n is the input images, and $Y_n = \{y_p^{(n)}, p = 1, \ldots, |I_n|\}$ is the predicted labels. $y_p^{(n)} \in \{0, 1\}$, which is defined in Equation (1). Then the loss function is defined as follows:

$$\mathcal{L}(\boldsymbol{W}) = -\beta \sum_{p \in Y_+} logPr(y_p = 1|I; \boldsymbol{W}) - (1-\beta) \sum_{p \in Y_-} logPr(y_p = 0|I; \boldsymbol{W}), \tag{4}$$

where $\boldsymbol{W}$ denotes the learning parameters of the network model and β is used to handle the imbalance of the background and foreground pixels numbers. Here, $\beta = |Y_-|/|Y|$ and $1 - \beta = |Y_+|/|Y|$. Y_+ and Y_- denote the foreground and the background of the label mask Y, respectively. The probability $Pr(\cdot)$ is computed by using a sigmoid function $\sigma(\cdot)$ on the final activation layer.

C. IMPLEMENTATION AND TRAINING DETAILS

Table 1 summarized the detailed configuration of the proposed multiscale fully convolution network which have been presented in Fig. 3. Here "conv" denotes the convolution operation, "max-pool" denotes the max pooling operation, "avgPool" denotes the average pooling operation and "deconv" denotes the deconvolution operation. The input layer corresponds to the input frames, and the output layer is the probability maps have the same size as the input.

Our MFCN model is implemented in TensorFlow [36]. The layers from the VGG-16 are initialized with the pre-trained weights [5], while other weights are initialized randomly with a truncated normal distribution $\mathcal{N}(0, 0.01)$. The AdamOptimizer method is used for updating our model parameters with a learning rate of 10^{-4}. During the training stage, the training data are augmented with horizontal flipping. Each category

TABLE 1. Detailed configuration of the proposed multiscale fully convolution network for background subtraction.

name	kernel	stride	pad	output size
input image	-	-	-	288*288*3
conv1-1	3*3	1	Yes	288*288*64
conv1-2	3*3	1	Yes	288*288*64
max-pool1	2*2	2	No	144*144*64
conv2-1	3*3	1	Yes	144*144*128
conv2-2	3*3	1	Yes	144*144*128
max-pool2	2*2	2	No	72*72*128
conv3-1	3*3	1	Yes	72*72*256
conv3-2	3*3	1	Yes	72*72*256
conv3-3	3*3	1	Yes	72*72*256
max-pool3	2*2	2	No	36*36*256
conv4-1	3*3	1	Yes	36*36*512
conv4-2	3*3	1	Yes	36*36*512
conv4-3	3*3	1	Yes	36*36*512
max-pool4	2*2	2	No	18*18*512
conv5-1	3*3	1	Yes	18*18*512
conv5-2	3*3	1	Yes	18*18*512
conv5-3	3*3	1	Yes	18*18*512
max-pool5	2*2	2	No	9*9*512
conv-X1	3*3	1	Yes	144*144*128
conv-X2	3*3	1	Yes	72*72*128
conv-X3	3*3	1	Yes	36*36*128
conv-X4	3*3	1	Yes	18*18*128
conv-X5	3*3	1	Yes	9*9*128
avgPool-C1	3*3	1	Yes	144*144*128
avgPool-C2	3*3	1	Yes	72*72*128
avgPool-C3	3*3	1	Yes	36*36*128
avgPool-C4	3*3	1	Yes	18*18*128
avgPool-C5	3*3	1	Yes	9*9*128
deconv-D5	3*3	2	Yes	18*18*128
deconv-D4	3*3	2	Yes	36*36*256
deconv-D3	3*3	2	Yes	72*72*384
deconv-D2	3*3	2	Yes	144*144*512
deconv-D1	3*3	2	Yes	288*288*640
output	1*1	1	No	288*288*2

video is trained for 20 epochs with a batch size of 5 frames. It takes about 17 hours to finish the whole training process with an NVIDIA Titan Xp GPU for the 53 sequences (each sequence is about 20mins) from the CDnet 2014 dataset and about 2 hours for the whole training frames from the SBM-RBGD dataset.

IV. EXPERIMENTAL RESULTS

This section describes the details about the evaluation dataset and metrics and presents the quantitative and qualitative results of the proposed algorithm.

A. EVALUATION DATASETS

As we mentioned early, we use two different datasets to evaluate the proposed algorithm. There are certain requirements for finding proper datasets for algorithm evaluation, especially in the research area of deep learning. The most important, a large number of labeled samples is necessary for training the network models. And the video sequences must span across a wide variety of categories under different challenges to make an exhaustive evaluation of the algorithm's capabilities. In this paper, we evaluate the proposed MFCN-based background subtraction method on the CDnet 2014 dataset [33] and the SBM-RGBD dataset [37] provided for the Change Detection Challenge, which with the goal of allowing performance comparison for recent and future background subtraction methods.

TABLE 2. Overview of the CDnet 2014 dataset.

Category	Videos	Total Frames	Evaluation Frames
baseline	4	6049	4413
cameraJ	4	6420	3134
dynamic	6	18871	13276
intermittent	6	18650	12111
shadow	6	16949	14105
thermal	5	21100	18055
badWeather	4	20900	17904
lowFramerate	4	9400	5204
nightVideos	6	16609	12285
PTZ	4	8630	5534
turbulence	4	15700	12204
Total	53	159278	121721

1) CDnet 2014 DATASET

The CDnet 2014 dataset consists of 53 videos from realistic scenarios with nearly 160 000 frames. Accurate human constructed ground truths are available for all sequences. These sequences are grouped into 11 categories namely: ''baseline'', ''camera jitter'', ''dynamic background'', ''intermittent object motion'', ''shadow'', ''thermal'', ''bad weather'', ''low framerate'', ''night videos'', ''pan-tilt-zoom'' and ''turbulence''. This is currently the most complete dataset for background subtraction. Among them, the first six sequences constitute the CDnet 2012 dataset. A complete overview of this dataset is depicted in Table 2. And Fig. 4 shows some sample frames and their corresponding ground truths. Over the years numerous background subtraction methods have been evaluated on this dataset and their quantitative results are published on the website. Therefore, in this paper, we use the CDnet 2014 dataset as our primary evaluation dataset.

2) SBM-RGBD DATASET

The SBM-RGBD dataset [37] provides all facilities for the SBM-RGBD Challenge, a set of ground-truthed synchronized color and depth sequences acquired by RGBD sensors. This dataset consists of 33 videos (nearly 15 000 frames) captured in video surveillance and smart environment scenarios. These videos span seven categories: ''bootstrapping'', ''color camouflage'', ''depth camouflage'', ''illumination changes'', ''intermittent motion'', ''out of sensor range'', and ''shadows''. All sequences come with pixel-wise ground truth labels and quantitative evaluation is made across whole sequences. However, be different with the CDnet 2014 dataset, for each sequence, only a few random frames' ground truths are publicly available on the website (about 1080 frames in the whole ∼15 000 frames).

B. EVALUATION METRICS

In order to make an exhaustive competitive comparison between different background subtraction methods,

FIGURE 4. The CDnet 2014 dataset [33]: The first row shows an original frame from each category and the second row shows its ground truth. From left to right: "baseline", "camera jitter", "intermittent object motion", "thermal", "pan-tilt-zoom" and "bad weather".

seven different performance metrics have been defined in [33] and [37]: Let *TP* stands for true positives and represents the number of correctly classified foreground pixels, *TN* stands for true negatives and represents the number of correctly classified background pixels, *FN* stands for false negatives and represents the number of incorrectly classified foreground pixels, and *FP* stands for false positives and represents the number of incorrectly classified background pixels. The seven metrics are defined as follows: Recall (*Re*), Specificity (*Sp*), False positive rate (*FPR*), False negative rate (*FNR*), Percentage of wrong classifications (*PWC*), Precision (*Pr*), F-Measure (*FM*).

- Recall $(Re) = \frac{TP}{TP+FN}$
- Specificity $(Sp) = \frac{TN}{TN+FP}$
- False positive rate $(FPR) = \frac{FP}{FP+TN}$
- False negative rate $(FNR) = \frac{FN}{TP+FN}$
- Percentage of wrong classifications $(PWC) = 100 \cdot \frac{FN+FP}{TP+FN+FP+TN}$
- Precision $(Pr) = \frac{TP}{TP+FP}$
- F-Measure $(FM) = 2 \cdot \frac{Re \cdot Pr}{Re+Pr}$

With the standardized evaluation tool provided by [33], we can easily compare our method with other state-of-the-art methods based on these metrics. For PWC, FNR and FPR metrics, lower values indicate higher accuracy, while for Recall, Specificity, Precision and F-Measure metrics, higher values indicate better performance. Among these metrics, we are especially interested in the F-Measure (*FM*) metric, which is commonly accepted as a good indicator of the overall performance of the background subtraction methods. Generally, if a method has high Recall scores without sacrificing Precision, that's a good algorithm. The F-Measure metric represents a balance between the Recall and Precision. As shown in [33], most state-of-the-art methods typically exhibit higher F-Measure scores than the worse performing methods.

C. POST PROCESSING

During the foreground detection stage, the foreground probability map is obtained with a softmax operation on the last score layer. Then the threshold is applied to the probability map and achieve the final binary mask. We test the threshold value from 0 to 0.6 to find the optimal for our method. Fig. 5 shows how the F-Measure scores vary with different threshold values on the CDnet 2014 dataset. We can see that a threshold with 0.05 gives the best performance.

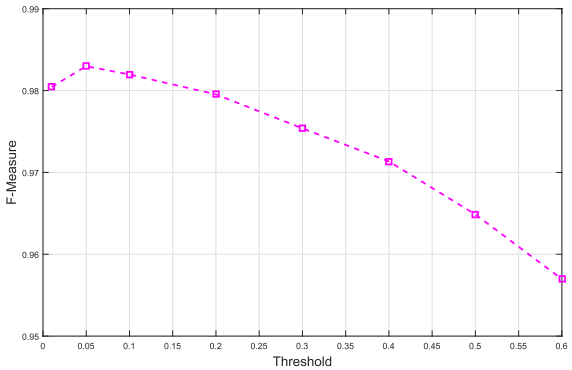


FIGURE 5. F-Measure scores of MFCN evaluated with different threshold values.

TABLE 3. Overall results in F-Measure with different sizes of median filter for post processing on the CDnet 2014 dataset.

	None	3×3	5×5	7×7	9×9	11×11
FM	0.9817	0.9824	0.9830	0.9821	0.9790	0.9754

Since the classify decision is made independently for each pixel. The foreground segmentation result can be benefited from regularization step, which combines information from neighboring pixels and assigns homogeneous labels on uniform regions. In our method, a simple median filter is used to enhance the spatial coherency and reduce the noises. Table 3 presents results with different median filter sizes. It can be seen that strong median filtering leads to higher F-Measure but it is also important to note that it also increases computational complexity. In this paper, we use a 5×5 median filter for all our experiments.

D. EXPERIMENTS ON THE CDNET 2014 DATASET

1) QUANTITATIVE EVALUATION

Firstly, to demonstrate our key contribution, the superiority of the proposed multiscale fully convolutional network

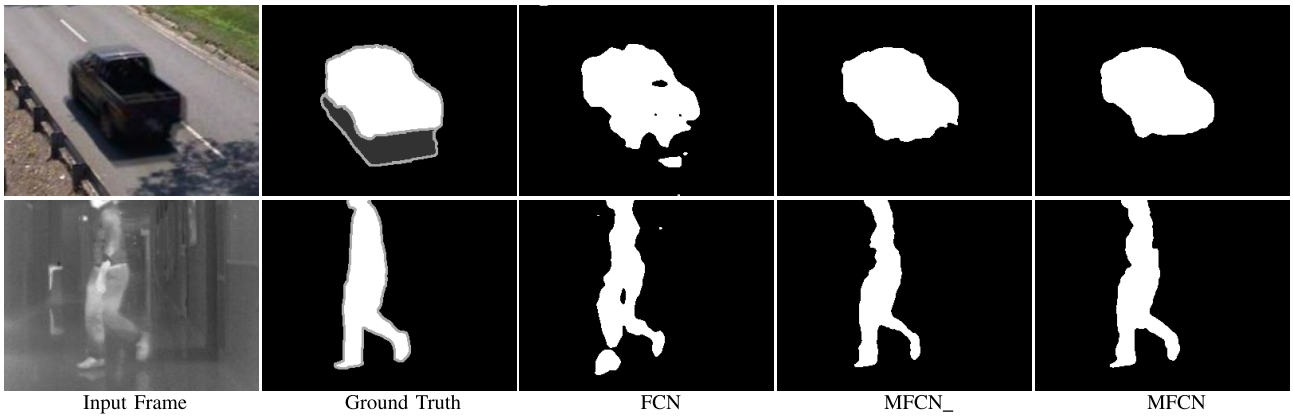


FIGURE 6. Visual comparison of foreground object detection results with different network architectures. (a) Input frame. (b) Ground Truth. (c) Detection results of FCN architecture. (d) Detection results of MFCN_ architecture. (e) Detection results of MFCN architecture.

TABLE 4. Average performance comparison of different network architecture on the CDnet 2014 dataset.

Network	Recall	Precision	F-Measure
FCN	0.8848	0.9178	0.8975
MFCN_	0.9462	**0.9881**	0.9662
MFCN	**0.9828**	0.9841	**0.9830**

architecture, we implemented two another models. The first one is based on the FCN [8] architecture but with 2 class output. The second one (denoted as MFCN_) is based on the MFCN architecture without contrast layers ($C_1 \sim C_5$,). In Table 4, we present the results of performance comparison. Here, we can see that MFCN outperforms others on F-Measure with a large margin and the Recall, Precision, and F-Measure scores of FCN are much lower than MFCN_ and MFCN. For MFCN_, although its Precision is larger than MFCN, the Recall is too low, so its F-Measure is much lower than MFCN. In Fig. 6, we made a visual comparison of foreground object detection results with these different architectures on two sequences. We can notice that results produced by FCN are very coarse, which contain many holes (False Negative pixels) and unconnected regions. However, for MFCN_ and MFCN, benefiting from stepwise upsampling strategy to combine different scales features, the final detection results are much more accurate. From Fig. 6, we can also see that the main difference between MFCN_ and MFCN is foreground object boundaries. Without the contrast layers, the foreground object boundaries of MFCN_ are not well preserved, which made the final foreground results much thinner and less accurate.

Secondly, we present the detail performance evaluation results of our method in Table 5, seven metric scores are reported. We can see that our method achieves an over F-Measure of 0.9830 on the dataset. On CDnet 2012 dataset (the first six categories), which mainly deals with traditional challenges of background subtraction, the F-Measure scores are all more than 0.98. For the latter five new add categories from CDnet 2014, which include videos captured under outside snowy conditions, low framerate videos with wavering global lighting conditions, urban traffic surveillance videos captured at night with glare effect caused by car headlights, videos obtained with pan-tilt-zoom cameras and long distance thermal surveillance videos with air turbulence under high temperature environments, although these categories are much harder to deal with, the results show that in addition to the lowFramerate, the F-Measure of other categories are all more than 0.97. As demonstrated in [14], a method with a F-Measure above 0.94 and a PWC below 0.9, then the segmentation results may be considered almost as good as the ground truth, since a simple dilation (or erosion) of one (or two) pixel of the ground truth may result in the F-Measure drop from 1.0 to about 0.94. This again shows the efficiency of our method. The reason for the F-Measure of lowFramerate and nightVideos are little lower may be due to the fact that the training data from CDnet 2014 contain more noise than CDnet 2012. We know that the ground truth labels provided from CDnet 2014 contain many out of scope regions, and the pixel classes are not defined in these regions. However, in our training data preparation stage, we treat these pixels as background, which may affect the accuracy of the model trained. If all the training pixels are labeled in the ground truth, we may get higher accuracy.

Finally, we also compare our MFCN-based method with some classical and state-of-the-art background subtraction methods. Due to space limitations, we choose following nine methods: CascadeCNN [14], IUTIS-5 [38], SharedModel [39], DeepBS [13], WeSamBE [40], SuBSENSE [25], PAWCS [34], C-EFIC [41], GMM [18]. Among them, CascadeCNN and DeepBS are CNN-based methods. In Table 6 we give the detail per-category *F-Measure* comparisons. The results are from the online evaluation server.[1] (Note: our result can be visited at here.)[2] And for a specific category, if the method obtains the best performance, the corresponding

[1] http://jacarini.dinf.usherbrooke.ca/results2014/
[2] http://jacarini.dinf.usherbrooke.ca/results2014/497/

TABLE 5. Complete results obtained with the proposed method on the CDnet 2014 dataset.

Category	Recall	Specificity	FPR	FNR	PWC	Precision	F-Measure
baseline	0.9897	0.9999	0.0001	0.0102	0.0393	0.9965	0.9931
cameraJ	0.9918	0.9998	0.0002	0.0082	0.0491	0.9960	0.9939
dynamic	0.9923	0.9999	0.0001	0.0076	0.0098	0.9989	0.9956
intermittent	0.9665	0.9998	0.0001	0.0334	0.2024	0.9985	0.9822
shadow	0.9873	0.9998	0.0002	0.0126	0.0647	0.9949	0.9911
thermal	0.9780	0.9998	0.0002	0.0220	0.0965	0.9968	0.9873
badWeather	0.9860	0.9998	0.0002	0.0139	0.0431	0.9843	0.9852
lowFramerate	0.9846	0.9998	0.0002	0.0153	0.0383	0.9317	0.9550
nightVideos	0.9681	0.9994	0.0006	0.0319	0.1280	0.9722	0.9701
PTZ	0.9864	0.9998	0.0002	0.0136	0.0247	0.9859	0.9860
turbulence	0.9793	0.9998	0.0002	0.0206	0.0287	0.9688	0.9740
Overall	0.9828	0.9998	0.0002	0.0172	0.0659	0.9841	0.9830

TABLE 6. Overall and per-category F-Measure scores on the CDnet 2014 dataset by different methods.

Method	Overall	$F_{baseline}$	$F_{cam.jitt.}$	$F_{dyn.bg.}$	$F_{int.mot.}$	F_{shadow}	$F_{thermal}$	$F_{bad.wea.}$	$F_{low.fr.}$	F_{night}	F_{PTZ}	$F_{trubul.}$
MFCN	**0.9814**	**0.9931**	**0.9939**	**0.9956**	**0.9822**	**0.9911**	**0.9873**	**0.9881**	**0.9353**	**0.9764**	**0.9818**	**0.9709**
CascadeCNN [14]	0.9209	0.9786	0.9758	0.9658	0.8505	0.9593	0.8958	0.9431	0.8370	0.8965	0.9168	0.9108
IUTIS-5 [38]	0.7717	0.9567	0.8332	0.8902	0.7296	0.8766	0.8303	0.8248	0.7743	0.5290	0.4282	0.7836
SharedModel [39]	0.7474	0.9522	0.8141	0.8222	0.6727	0.8898	0.8319	0.8480	0.7286	0.5419	0.3860	0.7339
DeepBS [13]	0.7458	0.9580	0.8990	0.8761	0.6098	0.9304	0.7583	0.8301	0.6002	0.5835	0.3133	0.8455
WeSamBE [40]	0.7446	0.9413	0.7976	0.7440	0.7392	0.8999	0.7962	0.8608	0.6602	0.5929	0.3844	0.7737
SuBSENSE [25]	0.7408	0.9503	0.8152	0.8177	0.6569	0.8986	0.8171	0.8619	0.6445	0.5599	0.3476	0.7792
PAWCS [34]	0.7403	0.9397	0.8137	0.8938	0.7764	0.8913	0.8324	0.8152	0.6588	0.4152	0.4615	0.6450
C-EFIC [41]	0.7307	0.9309	0.8248	0.5627	0.6229	0.8778	0.8349	0.7867	0.6806	0.6677	0.6207	0.6275
GMM [18]	0.5566	0.8382	0.5670	0.6328	0.5325	0.7322	0.6548	0.7406	0.5065	0.3960	0.1046	0.4169

F-Measure value is highlighted in bold. We can see that our method gets the highest F-Measure in all eleven categories, and the overall F-Measure score of our method is a big improvement compared with the current top method CascadeCNN. According to the statement of [13], current CNN-based background subtraction methods are scene specific, a model can only be used in a single scene. Thus the authors proposed to train a universal network with data from multiple scenes and achieve a universal model that can handle various scenes. It could be argued whether it is suitable for practical application. From the point of view of research, we also try to train a new model with all the training data from different sequences. The experimental results show that even using one model we also get the overall F-Measure of 0.9763 on the CDnet 2014 dataset, a little bit worse than the scene specific MFCN model, but a huge improvement than [13] of 0.7458. This fully demonstrates the effectiveness of our network architecture.

2) QUALITATIVE EVALUATION

To make a better visual comparison of the segmentation results under different challenges, we select the following sequences (without training frame): *highway* (815th) from the ''baseline'' category, *traffic* (1481th) from the ''camera jitter'' category, *sofa* (2019th) from the ''intermittent object motion'' category, *diningRoom* (3166th) from the ''thermal'' category, *turnpike_0_5fps* (1011th) from the ''low framerate'' category and *twoPositionPTZCam* (1041th) from the ''pan-tilt-zoom'' category. As shown in Fig. 7, the first column displays the input frames and the second column shows the corresponding ground truth. From the third column to the eighth column, the segmentation results of the following method are showed: our method (MFCN), CascadeCNN, IUTIS-5, DeepBS, SubSENSE and GMM. Visually, we can see that our results look better than all other methods, which show good agreement with the quantitative evaluation results. In the *highway* sequence which contains dynamic background (waving trees) and shadows, the segmentation results of our method are closest to the ground truth. In the *traffic* sequence with camera vibration challenge, the repetitive motions of the background objects resulting in many false positives are avoided in our method. And compared with other CNN-based methods CascadeCNN and DeepBS, which segment the foreground objects with several false negatives regions, our multiscale fully convolutional based method can learn more hierarchical features and segment the foreground object more accurately. *Sofa* sequence contains challenge about intermittent object motion. One man wearing a dark trouser and its color and texture is very similar to the sofa. In this case, even people are difficult to segment accurately, but our method also performs well in such environment with a concatenate and perfect foreground mask. The results of other method either include holes or divide foreground object into several parts. For the box left on the sofa which should be considered as foreground, we can observe that many methods absorb it into the background, resulting in many

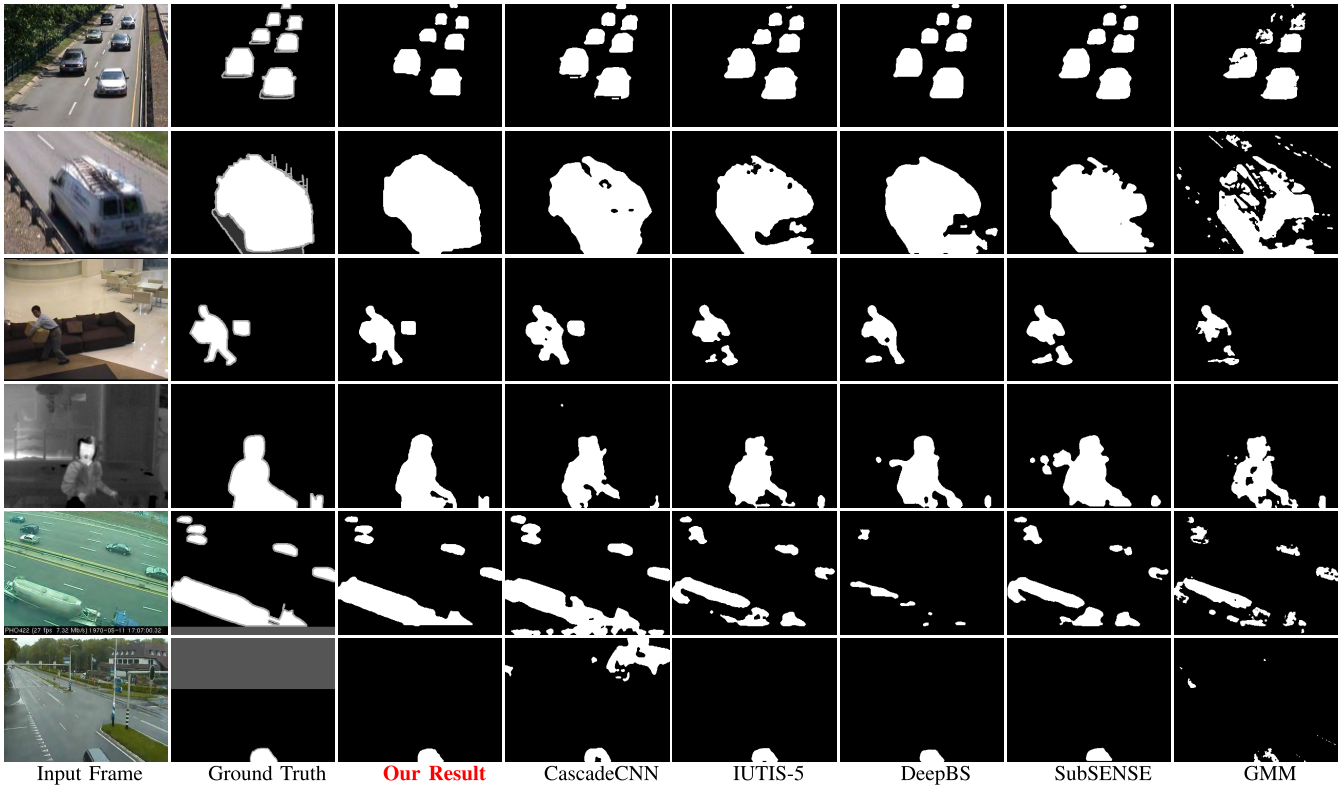


FIGURE 7. Qualitative performance comparison for various sequences (from top to bottom: ***highway, traffic, sofa, diningRoom, turnpike_0_5fps*** and ***twoPositionPTZCam***). The first column to the last column: input frame, ground truth, our segmentation result, CascadeCNN [14], IUTIS-5 [38], DeepBS [13], SubSENSE [25] and GMM [18] segmentation results.

false negatives while the proposed method can successfully detect the object. In the *diningRoom* sequence from the thermal category which contains infrared images with a narrow distribution range of pixel values. Although the foreground segmentation results of this frame are little difference with the ground truth, interesting, when we turned back and checked the original input frames, we found that from the human point of view, our segmentation results are more reasonable. The small region under the hand should not be considered as foreground. *Turnpike_0_5fps* sequence is obtained from low framerate videos. There may have a huge difference between adjacent frames. We can see that the foreground detected results from traditional methods are very noise. Results for our method are more close to the ground truth. In the *twoPositionPTZCam* sequence from the pan-tilt-zoom category, although the camera is not static, our method also detects the moving car perfectly. This again demonstrates the effectiveness of our method in difficult situations.

E. EXPERIMENTS ON THE SBM-RGBD DATASET

For the evaluation of the proposed method in the SBM-RGBD dataset, the following algorithms are compared: RGBD-SOBS [43] and RGB-SOBS [23], SCAD [42], SRPCA [45], and CwisarDH+ [44]. It should be noted that most of these methods exploit color and depth features for the background modeling, but the proposed method only uses RGB color information as well as the RGB-SOBS method. Since the whole ground truths are not available and the average provided ground truths frames for each sequence is about 30. It is difficult to train a model for each sequence due to the small mount of training data. So in the SBM-RGBD dataset, we only train one model with all training frames (about 1080 frames) from different sequences. Then the trained model is run across the whole dataset for foreground detection. Finally, we upload our results to make a quantitative comparison. In Table 7, we report average results for seven metrics on the whole dataset (the results are reported by the online evaluation server.)[3] For the *Ls_ds* and *TimeOfDay_ds* sequences in the dataset, there have no foreground objects through the whole duration, this leads to undefined values of Precision, Recall, and F-Measure metrics. So the average results are reported on the rest 31 sequences. From Table 7, we can see that the proposed method MFCN surpass all other methods in every metrics. Compared with CDnet 2014 dataset, the SBM-RGBD dataset contains stronger illumination changes, camouflage, and shadows. From the comparison between RGBD-SOBS and RGB-SOBS, we can see that a combination of color and depth information for foreground segmentation has a great improvement compared to the use of only color information. However, although

[3]http://rgbd2017.na.icar.cnr.it/SBM-RGBDchallengeResults.html

TABLE 7. Average results on the SBM-RGBD dataset.

Method	Recall	Specificity	FPR	FNR	PWC	Precision	F-Measure
MFCN	**0.9907**	**0.9981**	**0.0019**	**0.0093**	**0.2547**	**0.9807**	**0.9856**
SCAD [42]	0.9503	0.9914	0.0086	0.0497	1.1476	0.9323	0.9391
RGBD-SOBS [43]	0.9035	0.9949	0.0051	0.0965	1.2726	0.9473	0.9211
CwisarDH+ [44]	0.8434	0.9810	0.0190	0.1566	2.9010	0.8301	0.8254
SRPCA [45]	0.8611	0.9730	0.0270	0.1389	3.2551	0.8102	0.8210
RGB-SOBS [23]	0.8381	0.9763	0.0237	0.1619	4.7902	0.8125	0.7882

our algorithm only takes color information into consideration, it still achieves state-of-the-art performance with the F-Measure score of 0.9856. A new network architecture which combines color and depth features will be our future work.

F. PROCESSING SPEED

We know that background subtraction is often the first step in many high-level computer vision systems. Processing speed is the critical factor for researchers to be considered before choosing which method to use. During the detection stage, our MFCN method is run on the Ubuntu 16.04 operating system with an NVIDIA Titan Xp GPU, the average frame rate is nearly 20 fps(frames per second), which shows real time potential.

V. CONCLUSION

In this paper, we present a multiscale fully convolutional network (MFCN) architecture which takes advantage of different layer features for background subtraction. Benefitting from the deep features learned from different layers and the fully convolutional network architecture, our method achieves much higher foreground detection accuracy. And we also show that without building complex background models, the background subtraction process can be easily solved by the network classification. Experiments evaluated on the CDnet 2014 and SBM-RGBD dataset demonstrate that our method outperforms recent state-of-the-art background subtraction methods and has the potential for real-time applications.

ACKNOWLEDGMENT

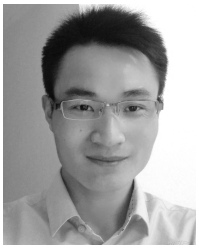
The authors would like to thank the anonymous reviewers for their helpful feedback.

They gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

REFERENCES

[1] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. Sci. Rev.*, vol. 11, pp. 31–66, May 2014.
[2] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.
[3] W. Kim and C. Jung, "Illumination-invariant background subtraction: Comparative review, models, and prospects," *IEEE Access*, vol. 5, pp. 8369–8384, 2017.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[5] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
[9] O. Ronneberger, P. Fischer, and T. Brox, "U-NET: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[10] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1265–1274.
[11] L. Wang, L. Wang, H. Lu, P. Zhang, and X. Ruan, "Saliency detection with recurrent fully convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 825–841.
[12] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. IEEE Int. Conf. Syst., Signals Image Process.*, May 2016, pp. 1–4.
[13] 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.
[14] 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.
[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
[16] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Dec. 2015, pp. 58–66.
[17] K. He, G. Gkioxari, P. Dollár, and R. Girshick. (2017). "Mask R-CNN." [Online]. Available: https://arxiv.org/abs/1703.06870
[18] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.
[19] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.
[20] D.-S. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.
[21] 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.
[22] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.
[23] L. Maddalena and A. Petrosino, "The SOBS algorithm: What are the limits?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 21–26.

[24] 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.
[25] 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.
[26] S. Javed, S. H. Oh, and S. K. Jung, "Foreground object detection via background modeling using histograms of oriented gradient," in *Proc. Int. Conf. Hum. Comput. Interaction*, Jan. 2013.
[27] D. Zhou and H. Zhang, "Modified GMM background modeling and optical flow for detection of moving objects," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 3. Oct. 2005, pp. 2224–2229.
[28] M. Heikkila and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.
[29] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," in *Proc. IEEE Winter Conf. Appl. Comput. Vision.*, Mar. 2014, pp. 509–515.
[30] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1301–1306.
[31] D. Zeng, M. Zhu, T. Zhou, and F. Xu, "An extended scale invariant local binary pattern for background subtraction," *IET Image Process.*, 2018.
[32] 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.
[33] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 387–394.
[34] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Universal background subtraction using word consensus models," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4768–4781, Oct. 2016.
[35] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1395–1403.
[36] M. Abadi *et al.* (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467
[37] M. Camplani, L. Maddalena, G. M. Alcover, A. Petrosino, and L. Salgado, "A benchmarking framework for background subtraction in RGBD videos," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 219–229.
[38] S. Bianco, G. Ciocca, and R. Schettini, "How far can you get by combining change detection algorithms?" [Online]. Available: https://arxiv.org/abs/1505.02921
[39] Y. Chen, J. Wang, and H. Lu, "Learning sharable models for robust background subtraction," in *Proc. IEEE Int. Conf. Multimultimedia Expo*, Jul. 2015, pp. 1–6.
[40] S. Jiang and X. Lu, "WeSamBE: A weight-sample-based method for background subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, 2017.
[41] G. Allebosch, D. Van Hamme, F. Deboeverie, P. Veelaert, and W. Philips, "C-EFIC: Color and edge based foreground background segmentation with interior classification," in *Proc. IEEE Int. Conf. Comput. Vis. Imag. Comput. Graph.*, Mar. 2015, pp. 433–454.
[42] T. Minematsu, A. Shimada, H. Uchiyama, and R.-I. Taniguchi, "Simple combination of appearance and depth for foreground segmentation," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 266–277.
[43] L. Maddalena and A. Petrosino, "Exploiting color and depth for background subtraction," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 254–265.
[44] M. De Gregorio and M. Giordano, "CwisarDH+: Background detection in RGBD videos by learning of weightless neural networks," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 242–253.
[45] S. Javed, T. Bouwmans, M. Sultana, and S. K. Jung, "Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA," in *Proc. Int. Conf. Image Anal. Process.*, 2017, pp. 230–241.



DONGDONG ZENG received the B.S. degrees in computer science and technology from Jilin University, Changchun, China, in 2013. He is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and physics, Chinese Academy of Sciences, China. His current research interests include object detection, tracking, and recognition.

MING ZHU is currently a Research Fellow and a Supervisor of Ph.D. Candidates of the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences. His research interests include digital image processing, television tracking, and automatic target recognition technology.