UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Real-time foreground object segmentation networks using long and short skip connections

Cong, L.; Zhang, S.; You, S.; Liu, X.; Zhu, Z.

DOI
10.1016/j.ins.2021.01.044

Publication date
2021

Document Version
Final published version

Published in
Information Sciences

License
CC BY

Link to publication

Citation for published version (APA):
Cong, L., Zhang, S., You, S., Liu, X., & Zhu, Z. (2021). Real-time foreground object segmentation networks using long and short skip connections. *Information Sciences*, *571*, 543-559. https://doi.org/10.1016/j.ins.2021.01.044

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:27 Jul 2022

Information Sciences 571 (2021) 543–559

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Real-time foreground object segmentation networks using long and short skip connections

Cong Lin [a], Shijie Zhang [b], Shaodi You [c,*], Xiaoxiang Liu [a], Zhiyu Zhu [d]

[a] School of Intelligent Systems Science and Engineering, Jinan University, Zhuhai, China
[b] Department of Pharmacology, School of Basic Medical Sciences, Tianjin Medical University, Tianjin, China
[c] Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands
[d] Electrical and Information Colleage, Jiangsu University of Science and Technology, Zhenjiang, China

ARTICLE INFO

Article history:
Received 10 May 2020
Received in revised form 11 January 2021
Accepted 13 January 2021
Available online 05 February 2021

Keywords:
Foreground segmentation
Lightweight model
Real-time inference
Surveillance

ABSTRACT

Foreground object segmentation is an important task with various applications in outdoor surveillance and navigation. Most existing methods focus on accuracy and therefore, are computationally expensive and low in speed, making them difficult to use in actual applications. In this study, we aim to address the issue of accuracy and efficiency trade-off. In particular, in contrast with existing methods that use fine-tuning routine on heavyweight pretrained models and/or optimization techniques to enhance results, we propose a lightweight end-to-end network that can be trained from scratch effectively and efficiently. First, long and short skip connections are used among convolutional blocks and within the bottleneck block. By doing so, information flow within the networks is enhanced during the training stage, and thus, the use rate of parameters in the model is increased, allowing a more compact and efficient network design. Second, we use feature fusions based on element-wise summing before each up-sampling layer to reduce the size of the decoder, accelerate the up-sampling process, and stabilize training convergence. Our proposed method is tested rigorously. In particular, we achieved 1000 times higher speed compared with state-of-the-art methods on CD2014 and SBI2015 datasets with comparable accuracy.
© 2021 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

In an intelligent surveillance system, detecting and automatically segmenting foreground objects are essential [1,2] because these techniques are used as preprocessing steps for a range of high-level tasks, e.g., human detection [3], traffic monitoring [4], event detection [5], visual tracking [6], magnetic medical imaging [7], and anomaly behavior detection [8]. In general, the problem setting assumes a relatively stationary background and a relatively stable camera. Therefore, a highly mobile object (typically a human or a vehicle) will occur with a significantly different motion and context, and thus, can be segmented from the background.

Segmenting foreground objects is not a trivial task because accuracy can be adversely affected by low light, blurred motion, camera jittering, non-rigid motion (particularly human motion), and similar foreground–background appearance.

* Corresponding author.
E-mail addresses: conglin@jnu.edu.cn (C. Lin), shijie.zhang@tmu.edu.cn (S. Zhang), s.you@uva.nl (S. You), tlxx@jnu.edu.cn (X. Liu), zzy@just.edu.cn (Z. Zhu).

https://doi.org/10.1016/j.ins.2021.01.044
0020-0255/© 2021 The Author(s). Published by Elsevier Inc.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Therefore, most existing rule-based and deep neural network-based methods focus on improving accuracy [9,1]. However, this condition hinders these methods from being run in real time and therefore, are far from real applications.

Our research considers accuracy and efficiency. We propose a lightweight network that uses long and short skip connections. The proposed network has two desired features: 1) The network is adaptive to view change. In contrast with many machine learning-based tasks whose model is trained once and used in many scenarios, the proposed network can be quickly trained from scratch and adaptive when the camera zooms or changes its angles. 2) Our proposed lightweight network does not require very large computational power or memory; it can be implemented on low-cost SoC, making it realistic in actual applications [10]. Designing a lightweight network for foreground segmentation is not an easy task because measuring the amount of visual information in a scene's background, which is the knowledge that should be embedded, is theoretically difficult. Thus, a small network may be unable to capture all the important features in a complex scene, resulting in unsatisfactory accuracy.

In particular, the proposed method employs a U-like net with long skip connections between encoder and decoder blocks and a short skip connection within the bottleneck. To achieve efficiency, we adopt the following strategy. The skip connections in the end-to-end network facilitate data flow and maximize the usage of parameters in the model, enabling a lightweight design and leading to faster networks. Moreover, computationally expensive components, e.g., batch normalization, sigmoid activation, are mostly avoided using such designs. We also adopt a computationally efficient loss function to improve training efficiency further. For accuracy, instead of adopting a fine-tuning pretrained routine that is prone to overfitting and eventually results in low accuracy [11]12, we train our model from scratch. In particular, we use thinner convolutional layers and data augmentation to help avoid overfitting.

The proposed method is rigorously tested on two popular datasets: Change Detection 2014 (CD2014) [13] and Scene Background Initialization 2015 (SB2015) [14]. The experimental results are statistically analyzed and compared with results produced by state-of-the-art methods.

The contributions of the proposed work are as follows:

- An efficient and accurate foreground segmentation solution is proposed.
- A novel lightweight end-to-end convolutional neural network (CNN) model is designed using long and short skip connections, which reduces model size and stabilizes convergence during training.
- The proposed model is tested rigorously on the CD2014 and SBI2015 datasets and compared with other state-of-the-art methods. We achieve comparable accuracy, while speed is 1000 times faster and model size is less than 0.1%.

Moreover, compared with existing solutions in the literature, the proposed model is a full end-to-end lightweight network that can be efficiently and effectively trained from scratch, making it a novel type of approach for this problem. We also experimentally demonstrate that some high-cost routines, e.g., fine-tuning a large pretrained backbone, preprocessing input data, and post-processing optimization for the results, are unnecessary. The remainder of this paper is presented as follows. Related studies are reviewed in Section 2. We introduce the problem formulation, the motivation, the network architecture, and the implementation details in Section 3. The experimental details and results are presented and discussed in Section 4. Finally, the study is concluded in Section 5.

2. Related works

Existing methods for the foreground segmentation problem can be classified into two categories: rule-based methods and CNN-based methods. Most rule-based methods use image processing and data mining techniques to construct and maintain a dynamic background. Foreground objects are extracted by comparing the dynamic background with new incoming frames. Although effective to a certain extent, these methods are not robust to illumination or content variations. Given that more data covering different types of variations are available at present, researchers are taking full advantage of the embedding capability of CNNs. Current advances are driven by CNN-based methods.

Rule-based Methods

Most of the early proposed solutions belong to the former category of background subtraction. Background subtraction-based methods [1] build a background model by using statistical theories and prior knowledge of existing video data. Thereafter, the background model is used to subtract the newly arriving frames. The subtracted differences between the arriving frames with a metric and the embedding information in the background model serve as important cues for segmenting foreground objects. A well-defined post-processing technique is typically used in refining the preliminary results from the subtraction. During a certain period, methods based on Gaussian mixture model (GMM) [15–17] are the mainstream solutions. An early work by Maddalena et al. [18] proposed to segment foreground objects by taking advantage of spatial coherence and incorporating it into background subtraction. The decision problem is optimized via fuzzy logic, which has long been used in image segmentation and edge detection [19–23], and the results are promising at that time. Thereafter, the research community has shifted its focus to exploring different types of data mining techniques to overcome challenges in background variations. Barnich et al. [9] proposed an efficient conventional method, called ViBe, setting the baseline to a higher level.

Vibe randomly samples multiple pixels at the same location or in the neighborhood along the past timeline and uses these pixels to construct a background model. The background model is adaptively updated concurrently with the background dynamics during online foreground segmentation. Hofmann et al. [24] adopted the framework of ViBe and developed an improved variant by introducing a self-adaptive threshold for foreground segmentation and a dynamic learning rate for background model updating. Zhang et al. [25] proposed quantitatively estimating dynamics in a video; the learning rates and the number of Gaussian components of GMM is adaptively updated in accordance with the estimated scene dynamics. St-Charles et al. [26] proposed SuBSENESE, which introduces spatiotemporal binary features and color information into estimating foreground changes. The parameters of the SuBSENESE model are dynamically adjusted using pixel-level feedback loops. Median filter-based post-processing is applied to remove false positive noise and refine the foreground map. Thereafter, the same authors proposed PAWCS [27] as an improvement to their previous work. PAWCS constructs a background model by encoding a dictionary model with color and local binary features. Jiang et al. [28] proposed WeSamBE, which is a weighted sample-based method. The background model in WeSamBE is in the form of a template library, and each template sample is assigned with a variable weight. To replace obsolete samples with new samples and update active samples with increasing weights, a reward-and-penalty weighting strategy is introduced to maintain the library. A low rank-based method was proposed by Zhang et al. [29]. This method considers appearance consistency and spatial compactness. Isik et al. [30] proposed SWCD, which maintains a dynamically updating background image by using a sliding window approach.

CNN-based Methods

CNN was introduced lately for solving the foreground segmentation problem. The earliest work was that of Lim et al. [31], who proposed an encoder-decoder architecture for segmenting foreground objects in stationary backgrounds. Foreground masks are implicitly inferred from the CNN model. The segmentation results are optimized with post-processing of superpixel fitting, hole filling, and false positive suppression. Wang et al. [32] proposed Cascade-CNN. Cascade-CNN consists of two cascaded CNN models; one model is used to predict coarse foreground masks, and the other model functions as an image refiner for the final foreground masks given the original images as prior information. DeepBS, proposed by Babaee et al. [33], builds an explicit background model in image form by combining the foreground mask from the results of the flux tensor algorithm [34] and the SuBSENSE algorithm [26]. The background image, which the CNN model conditions on, is used as prior information for inference. Lim et al. [11] proposed FgSegNet, which is related to SegNet [35] in semantic segmentation. FgSegNet uses ImageNet pretrained VGG-16 [36] as backbone encoder and average binary cross entropy as training loss. In the variant FgSegNet-M model, images are run through the pretrained model three times to obtain feature maps and then the decoder up-samples feature maps to form foreground maps, which are further filtered with a threshold into the final foreground mask. Zeng et al. [37] proprosed MFCN, which is generally similar to FgSegNet [11], i.e., both use VGG-based encoder [36] and a feature polling decoder. In contrast with FgSegNet [11], MFCN enhances feature flow between the encoder and the decoder by building multiple data pipes for shipping feature maps in different scales. BSUV-Net [38], proposed by Tezcan et al., is based on fully convolutional U-net [39]. In contrast with other CNN-based methods that are trained on a video to segment unseen frames from the same scene, BSUV-Net trains a model to segment different unseen videos. However, the performance for a specific video is considerably below those of other state-of-the-art methods. Similar to Cascade-CNN [32], BMN-BSN [40] adopts an architecture with two networks. The two networks are nested and trained together. One network is the scene feature extractor and uses a short sequence of frames as the input. The other network performs subtraction by using a target frame and the previous features.A pixel-wise detection method for moving foreground objects was proposed by Chen et al. [12]. This method stacks a recurrent neural network (RNN) and a conditional random field (CRF) model [41] at the tail of a deep model that uses VGG [36] or ResNet-50 [42] as backbone. The RNN model is used to infer object motion against the background, while CRF is used to optimize the results in the form of masks.

Considering the increasing amount of available data for the foreground segmentation problem and the limitations of rule-based methods in coping with a wide range of challenges, recent studies have shifted their focus to improving the utilization of the latest CNN architectures. However, current state-of-the-art CNN-based methods exhibit a least one of two drawbacks. 1) Without considering the amount of training data, an excessively large model that uses a large pretrained network as its backbone is adopted to encode less visual information disproportionately. SecondPhaseA large backbone network requires excessive computational resources and reduces speed in the training and inference pipeline. Examples include the cases in [12,11]. 2) Unnecessary preprocessing (e.g., case in [33]) and/or post-processing optimization (e.g., cases in [12,31,32]) are performed in the process pipeline, leading to overcomplex methods and increasing computational cost.

3. Real-time end-to-end foreground segmentation network using long and short skip connections

3.1. Motivation

Although some current state-of-the-art methods are effective in certain scenarios, their potential for widespread practical applications remain low due to prohibitive computational and memory costs. We have identified the two aforementioned drawbacks in current state-of-the-art methods in the literature and aim to overcome them in the present study.

One of the problems in these methods that leads to high cost is the false assumption that the amount of knowledge in a scene's background and its dynamics that a model is required to "remember" and generalize is in the same scale as those in

generic image classification. On the basis of this false assumption, methods, such as those in [11,12], adopt a CNN model proposed for the classification task in a huge ImageNet dataset as the backbone network. That is, the capacity of the image classification model overkills the foreground segmentation, resulting in a serious waste of computational resources.

Another trade-off is the use of data preprocessing techniques and post-processing optimization in the pipeline. Some methods fail to identify the powerful regression capability of CNN-based models. Given sufficient data, a trained model is capable of inferring data from one domain to another in a direct end-to-end manner. By contrast, post-processing optimization makes key components heterogeneous and divides the pipeline into different stages [32], resulting in unnecessary overhead and reduced speed. In this study, we intend to demonstrate that data preprocessing techniques and post-processing refinement for the results, neither of which is part of the CNN architecture, are unnecessary for inference in foreground segmentation.

In summary, the motivation is to address the aforementioned problems by proposing a lightweight end-to-end CNN model that can be trained from scatch for foreground segmentation.

3.2. Problem formulation

To design an efficient and effective foreground segmentation model, we adopt an end-to-end approach and focus on enhancing the performance of the end-to-end model. An end-to-end network [43–45] [46] is a variant of CNN that infers signals from one domain directly into another domain without requiring complex preprocessing and post-processing optimization. One of the key advantages of end-to-end models is the simplicity of the solution pipeline in which preprocessing and post-processing computation are no longer required. In our task, the proposed end-to-end networks takes in three-channel RGB image sequence, which contain a background scene and possible foreground objects, as input on one end. The end-to-end networks directly output the predicted corresponding foreground mask sequence on the other end. To segment irregularly shaped foreground objects, a foreground mask is commonly used to determine whether the corresponding pixels belong to the foreground or background. The motivations for basing our proposed model on an end-to-end framework are straight-forward. 1) The setting of one input end and one output end saves computational resources and improve efficiency. 2) The simple form and relative symmetric configuration of the network make implementation easy in applications; 3) The model fits perfectly into the input-output setting of the foreground segmentation problem.

Let the input image, output probability map, predicted foreground mask, and its groundtruth foreground mask be $I_{in}, M_{out}, M_{pred}$, and M_{gt}, respectively. The input image I_{in} is a three-channel RGB image with size $N \times N \times 3$, where N is the height and width of the input image, and 3 is the number of channels. The M_{out}, M_{pred}, and M_{gt} are of only one channel, i.e., $N \times N$. The value in probability map M_{out} is within the range of $[0, 1]$, indicating the likelihood that the model will consider the corresponding pixel to belongs to the foreground. Meanwhile, the values in the predicted foreground mask M_{pred} and groundtruth foreground mask M_{gt} are either 0 or 1, which respectively labels the pixel as the background or the foreground. In the inference stage, image I_{in} is inputted into the end-to-end foreground segmentation model $\mathscr{F}$ parameterized by θ, and the model outputs probability map M_{out} via inference.

$$M_{out} = \mathscr{F}(I_{in}|\theta) \tag{1}$$

In the training stage, loss is computed by measuring the difference between the probability map M_{out} and the corresponding groundtruth mask M_{gt} by using preset metric. Then, the loss value is back-propagated to update the filter weights in the networks. The goal is to train a model to output correct foreground object masks, i.e., to search for an optimal set of parameters for the model that can output the probability map consistently with the groundtruth mask. Given a defined loss metric $\mathscr{L}$, the optimization can be expressed as follows:

$$\widetilde{\theta} = \arg\min_{\theta} \mathscr{L}\big(\mathscr{F}(I_{in}|\theta), M_{gt}\big). \tag{2}$$

Once $\widetilde{\theta}$ is determined, the optimal model is used to infer new incoming frames. In the testing stage, the final predicted foreground mask M_{pred} is obtained by thresholding probability map M_{out} with a preset value τ:

$$M_{pred} = \mathscr{F}\Big([I_{in}, \tau]|\widetilde{\theta}\Big). \tag{3}$$

During inference within $\mathscr{F}$, given the preset threshold τ and an output probability map M_{out} from the neural networks, the thresholding process is expressed as follows:

$$M_{pred}^{(x,y)} = \begin{cases} 0 & for\ M_{out}^{(x,y)} < \tau \\ 1 & for\ M_{out}^{(x,y)} \geqslant \tau \end{cases}, \tag{4}$$

where x and y are the horizontal and vertical locations of a pixel, respectively, in the corresponding maps. The predicted foreground mask M_{pred} is highly related to the probability map M_{out}. Thus, to train the model to be robust to different environmental variations, designing and training an end-to-end $\mathscr{F} : (I_{in}, \hat{n}) \rightarrow M_{out}$ are preferable, where $\hat{n}$ denotes noises or environmental variations, such that the difference between M_{out} and M_{gt} is minimal. The procedure for training and inference is illustrated in Fig. 1.

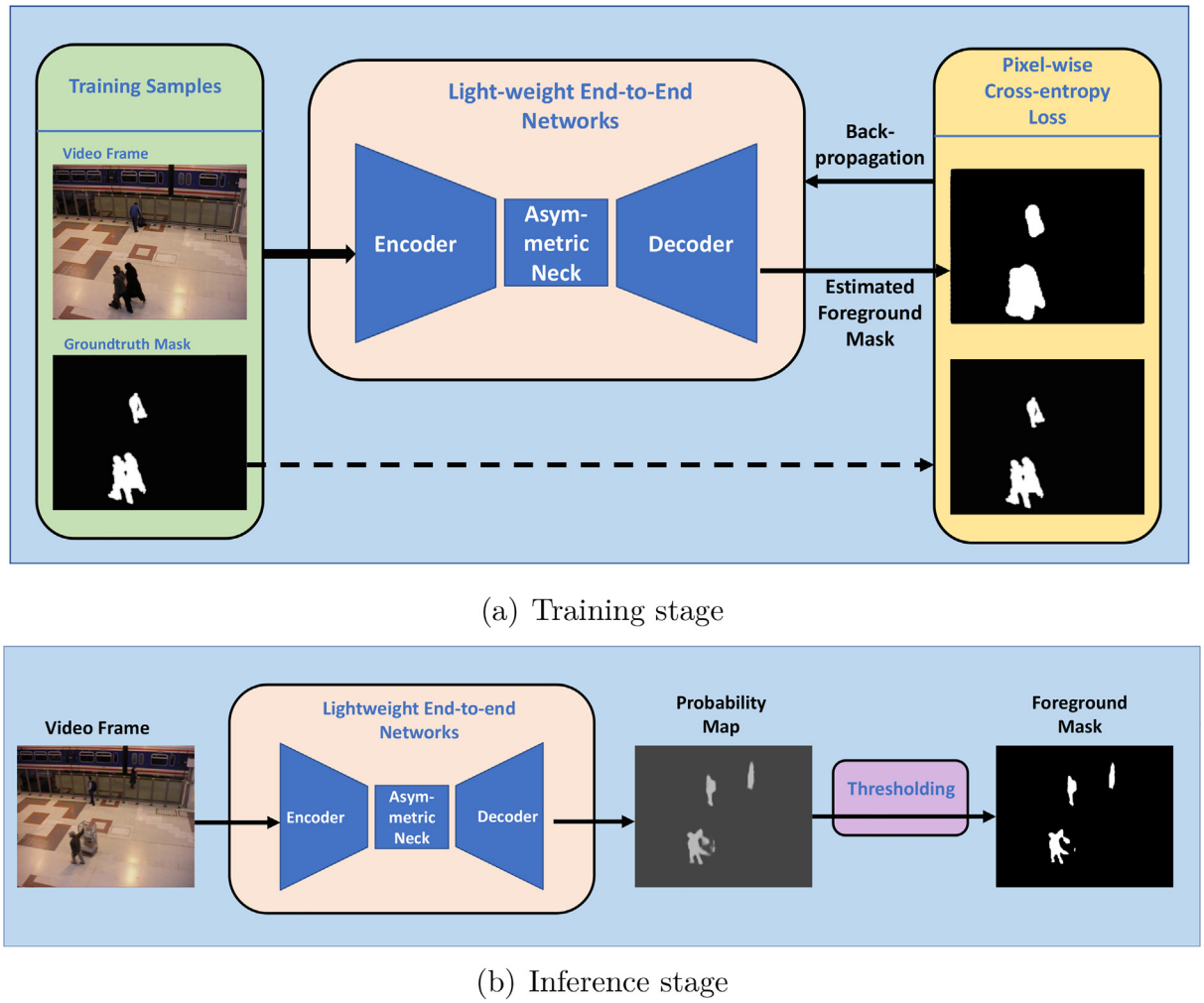


Fig. 1. Training and inference frameworks of the proposed method.

3.3. Proposed lightweight end-to-end architecture

The design of the proposed novel lightweight foreground segmentation model involves two steps. 1) A scalable model in relative symmetric encode-decode form is proposed accordingly. 2) The optimal hyperparameters for the scalable model are determined via a grid search. The proposed foreground segmentation model exhibits an encoder-decoder structure with long and short skip connections. The scalable architecture can be divided into three parts: the encoder, the decoder, and the bottleneck block. The abstract structure of the three parts in the model is shown in Fig. 1, and the detailed architecture of the proposed networks is presented in Fig. 2.

3.3.1. Design of proposed architecture

SecondPhaseTo achieve lightweightness, the overall design pattern keeps the layers near the two ends as shallow as possible and attempts to fully utilize the encoding capability of the layers/blocks deep down the bottleneck. We avoid using dropout in all the layers of the network because dropout will slow down the training process.

The encoder consists of N layers, and each layer down-samples the features from the previous layer. The size of all the filters is 4×4 with a stride of 2. To increase flexibility, biases are also used for each filter. Given an index $i \in \{1, 2, \ldots, N\}$ for a layer in the encoder of N convolutional layers, the number of filters in the corresponding down-sampling layers is $K = \ell * \ 2^{\lceil i/2 \rceil - 1}$. We avoid increasing the number of filters in the beginning layers, such that the intermediate features remain small, helping keep the feature maps reasonably small and reducing follow-up cost.

The bottleneck component, called asymmetric bottleneck and illustrated in Fig. 3, consists of bifurcate data flow paths. One path goes through the four mini-blocks in the bottleneck. Each mini-block is an extremely lightweight operation block that further consists of one convolutional layer, one instance normalization, and one parametric ReLU layer. The cover sizes

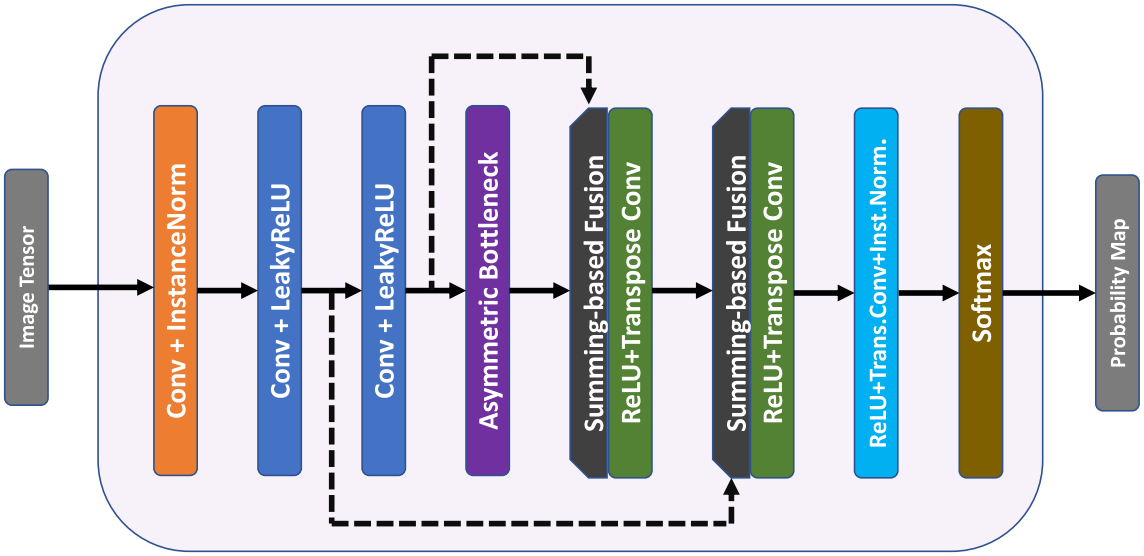


Fig. 2. Architecture of the proposed lightweight end-to-end network.

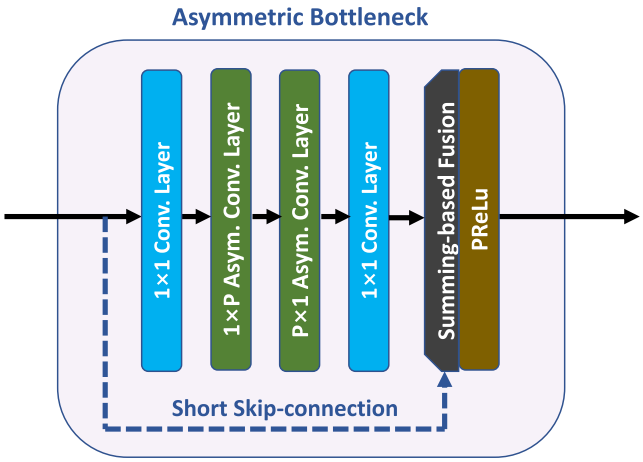


Fig. 3. Asymmetric bottleneck of the proposed lightweight end-to-end network.

of these convolutional filters in the mini-blocks are 1×1,$1 \times P$,$P \times 1$, and 1×1. The internal layer with a filter size of $P \times 1$ or $1 \times P$ is an asymmetric convolutional layer that rapidly executes and generalizes spatial information in the vertical or horizontal direction, respectively. P, which determines the sizes of the filters, is related to the size of deep feature maps that reach the bottleneck. The sizes of these deep feature maps are related to depth, i.e., N, in accordance with the definition of the scalable encoder. As the encoder goes deeper, the size of the deepest feature maps becomes smaller. Therefore, a smaller P for the asymmetric convolutionanl layer is required for the bottleneck component. Assuming that $N \leqslant 5$ and based on the output-input between the encoder and the bottleneck, the relation between P and N is defined as $P = 1 + 2^{(5-N)}$.The feature maps in the bottleneck are normalized instance-wise after passing through each convolutional layer. Before the feature maps are outputted from the bottleneck, a summing-based feature fusion is performed before a PReLU activation layer.

The decoder mirrors the encoder. The input feature maps are fused via element-wise addition, and the deconvolutional layers are employed in up-samplings. Given that the index of the convolutional layers is counted from deeper layers, the K for the layers in the decoder is $K = \ell * \ 2^{\lfloor (N-i)/2 \rfloor}$.Before the final output, a softmax layer is used to normalize the mask values within the range of $[0, 1]$.

3.3.2. Summing-based feature fusion

All the data fusions in the networks are summing-based feature fusion, which is the element-wise addition from one feature map to another. Summing-based data fusion exhibits two advantages over concatenation-based fusion in conventional

U-net [39]. 1) It reduces the size and computational cost of the decoder. 2) It fully utilizes encoder and decoder embedding capacities. We reason that the symmetric form of the end-to-end network fully use encoder and decoder, and the summing-based feature fusion is method for making the end-to-end network more symmetric. In our task, the dimension of the final output of the network is merely 1/3 of the input data and the output contains information in a considerably simpler form, i.e., an object mask. To correctly output a data map that is 1/3 the size of the input, the 2× feature map in the decoder is too redundant, similar to the case in conventional U-net [39]. Assuming that each skip connection and up-sampling provide approximately 1/2 useful deep features for the final results is reasonable. Therefore, designing a decoder with the same size as the encoder is appropriate, and the summing-based data fusion fits this requirement. Moreover, summing-based data fusion cuts the computational cost and model size of the decoder into half. In the experiments, we found that the concatenation of intermediate feature maps from the encoder leads to an unstable loss curve and more dramatic training convergence.

3.3.3. Long and short skip connections

Inspired by [47,48], long and short skip connections are used in the proposed architecture to facilitate data flow and enhance the usage rate of the parameters. To train a model thoroughly and fully utilize encoding capability in the bottleneck block, we keep bottleneck layer as compact as possible and prevent it from becoming too deep. In the foreground segmentation task, the output only requires rough spatial information from deeper layers. Thus, we design a smaller bottleneck layer that accelerates data passage and generates sufficient spatial features. Compared with stationary background subtraction methods or dynamic background models, such as GMM [15–17], the U-net-like model is capable of fitting the dynamic information of scene background into convolutional layers with large-scale parameters. Compared with other tasks, such as super-resolution and style transfer, our output is not texture-rich but rather a mask. Thus, the network is not required to be excessively deep and the bottleneck should be compact and fast to infer coarse spatial features. Following the experimental discovery in [47] and to better use all the parameters in the deepest layers, we use a short connection in the bottlenet block in the architecture, helping enhance the utilization rate and further ensuring compactness and efficiency.

The long skip connections pass the intermediate feature maps from the ith layer of the encoder to the last ith layer of the decoder directly and in parallel. In [31], which used conventional encoder-decoder networks, the information passing through the bottleneck layers is not abundant when the input and feature maps go through down-sampling layers in the decoder because the features describing texture details are encoded in early layers. In the reverse process, i.e., up-sampling in the decoder, the deconvolutional layers may have insufficient features for effectively taking account of most of the details required for inferring the foreground object mask. Using long skip connections, which ship low-level features directly across networks, can overcome this drawback. A long skip connection sets up a pipeline for sharing low-level features from encoding layers to the corresponding decoding layers. Compared with FgSegNet-M [11], which is required to rescale input images and go through feature extraction networks three times, using long skip connections allow richer information to pass to the other end, making one pass sufficient for effectively inferring the foreground mask.

3.3.4. Grid search and optimal solution

The optimal solution is determined via a grid search. A grid search for the key hyperparameters N and ℓ, which respectively determine the depth and scale of the model, is performed. In SubSection 4.3, our ablative studies found the optimal hyperparameters for the scalable model that balances efficiency and accuracy via grid search. Mathematically, Eq. 3 that searches for the optimal model can be further extended and expressed as

$$\widetilde{\theta} = \arg\min_{\theta,\langle N,\ell\rangle} \mathscr{L}\left(\mathscr{F}\left(I_{in}|\theta^{\langle N,\ell\rangle}\right), M_{gt}\right). \tag{5}$$

Empirically, we found that $N = 3$ and $\ell = 32$ are the optimal and balancing settings for the solution. Detailed configurations of the optimal solution are presented in Table 1. Experimental studies on the effectiveness of skip connections are also provided in the same subsection.

3.4. Implementation details

The proposed method is implemented on PyTorch 0.4.1. The implemented method is trained and tested on an X86 PC powered by Intel i5 CPU@3.7 GHz, 16GB RAM, Ubuntu 16.04 OS, Nvidia GTX Titan Xp with 12 GB graphic memory. Considering that the input and output sizes of the end-to-end networks are 256 × 256, images of different sizes and scale ratios are rescaled to the input size. The last layer is a softmax layer, and the logits are normalized; thus, the direct output of the network is a probability map with a pixel value range of [0,1], and the corresponding final foreground mask is generated by thresholding the output probability map. In all the experiments, the threshold value τ is fixed at 0.5, which is commonly considered the default for softmax output. Empirically, we found that once the model is well-trained, the output probability maps will be polarized and overall performance will be insensitive to the preset thresholding value. To simulate the illuminational variance and synthetize more variants of the training samples, data augmentation is used during training. The RGB image samples are augmented by randomly jittering brightness, contrast, saturation, and hue. The range of color jittering is 10%, 10%, 10%, and 2% for brightness, contrast, saturation, and hue, respectively. Since tiny foreground objects are considered in some challenges of the evaluation dataset, we avoid using any geometric transforms, e.g., distortions and translations.

Table 1
Detailed configurations of the proposed lightweight end-to-end networks in optimal setting ($\langle N = 3, \ell = 16 \rangle$, which is obtained from the ablative study in Section 4.3).

Layer Name	Filter Settings	Number of Parameters	Size of Output	Comments
Input	–	–	$256 \times 256 \times 3$	(RGB image)
Enc_1	$4 \times 4 \times 32$	$4 \times 4 \times 3 \times 32 + 32$	$128 \times 128 \times 32$	stride = 2; use bias
Enc_2	$4 \times 4 \times 32$	$4 \times 4 \times 32 \times 32 + 32$	$64 \times 64 \times 32$	stride = 2; use bias
Enc_3	$4 \times 4 \times 64$	$4 \times 4 \times 32 \times 64 + 64$	$32 \times 32 \times 64$	stride = 2; use bias
Neck_1	$1 \times 1 \times 16$	$1 \times 1 \times 64 \times 16$	$32 \times 32 \times 64$	stride = 1; padding = 0; no bias
Neck_2	$1 \times 5 \times 16$	$1 \times 5 \times 16 \times 16$	$32 \times 32 \times 16$	stride = 1; padding=(2,0); no bias
Neck_3	$5 \times 1 \times 16$	$5 \times 1 \times 16 \times 16$	$32 \times 32 \times 16$	stride = 1; padding=(2,0); no bias
Neck_4	$1 \times \times 1 \times 64$	$1 \times 1 \times 16 \times 64$	$32 \times 32 \times 64$	stride = 1; padding = 0; no bias
Summing	–	–	–	copy of feature map from Neck_1
PReLU	–	–	–	–
Summing	–	–	$32 \times 32 \times 64$	copy of feature map from Enc_3
Dec_1	$4 \times 4 \times 64$	$4 \times 4 \times 64 \times 32 + 32$	$64 \times 64 \times 32$	stride = 2; use bias
Summing	–	–	$64 \times 64 \times 32$	copy of feature map from Enc_2
Dec_2	$4 \times 4 \times 32$	$4 \times 4 \times 32 \times 32 + 32$	$64 \times 64 \times 32$	stride = 2; use bias
Summing	–	–	$128 \times 128 \times 32$	copy of feature map from Enc_1
Dec_3	$4 \times 4 \times 32$	$4 \times 4 \times 32 \times 1 + 1$	$256 \times 256 \times 1$	stride = 2; use bias
Softmax	–	–	–	–
Output	–	–	$256 \times 256 \times 1$	(2D probability map)

Meanwhile, groundtruth masks are not augmented and kept constant while training. For a typical training set of 200 images, 300 epochs are generally sufficient for training the model to converge and produce satisfactory results. This training process only takes approximately 3 min, which is extremely fast for machine learning-based computer vision tasks. Even for the most challenging image sequences in the datasets, trainings converge within 5 min. Learning rate is initially set as 0.0001, and a scheme for dynamic learning rate is adopted. The model is trained using a constant learning rate for the first 20 epochs. In the later 280 epochs, the learning rate decreases linearly and reaches 0 at the end of training. Loss is computed using an average binary cross-entropy loss for each pixel in a frame. Batch size is set as 1, which requires small memory space in each training iteration. In contrast with the model from [11] that requires training with a shuffle sampling scheme, our model does not need sampling data through a shuffling approach and can be trained in sequential sampling and random shuffling, making the proposed method more flexible to use in streaming video scenarios.

4. Experimental results

The proposed method is tested on CD2014 [13] and SBI2015 [14] datasets. The CD2014 dataset is thus far the largest dataset for the foreground segmentation problem. It contains different challenges that cover most daily scenarios in the real world: 53 image sequences (decomposed from videos) divided into 11 categories, including baseline, night videos, thermal videos, and dynamic background. The SBI2015 dataset provides additional 14 image sequences in various challenges, and all image samples come with corresponding pixel-level groundtruths. The two datasets constitute a comprehensive evaluating data environment for foreground segmentation methods. Some samples from the CD2014 dataset are shown in Fig. 4. To avoid the potential of a supervised method remembering all the knowledge in a dataset, we split each dataset into training and test sets. To ensure fairness of comparison, datasets are divided following the scheme used in [11]. To prove the lightweightness of the proposed method, we also measure the computational costs by using various metrics. Except in the subsection on ablative study, the N and ℓ for the scalable model are always $N = 3$ and $\ell = 32$, which are the settings for the optimal solution. In the ablative study, we examine variants of the architectures and present their relative effectiveness through quantitative results.

To demonstrate the advantages of the proposed method in an objective and fair manner, we also compare its performance with state-of-the-art methods. The compared methods can be classified into classic methods and CNN-based methods. GMM-Spatial Info [16], SuBSENSE [26], PAWCS [27], ShareModel [49], IUTIS-5 [50], and Structured-GsMM [17] fall into the former catogory. Meanwhile, DeepBS [33], Cascade [32], FgSegNet-S [11], BMN-BSN [40], SFEN(VGG) [12], SFEN (ResNet50) [12], and BSUV-Net [38] are CNN-based methods. Among the classic GMM-based methods [16,49,17], Structured-GsMM [17] is the latest improved method. The current performance baseline, with single pass inference, is set by the CNN-based FgSegNet-S [11], which is the single pass version of FgSegNet-M [11]. In general, classic methods collect local information to construct a model, while CNN-based methods use image frames distributed across a video. Thus, CNN-based methods are expected to be more robust to various challenges. This expectation is met in our experiments. Our method is related to CNN theory but not based on the aforementioned methods.

4.1. Experimental analysis of the CD2014 dataset

For the experiments on the CD2014 dataset, 200 samples with their groundtruth masks are randomly selected from each image sequence to form a training set. These samples are not necessarily in consecutive frames or equally distributed across

(a) Highway (b) Canoe (c) LakeSide (d) Port-17fps (e) WetSnow

Fig. 4. Image samples with groundtruths from the CD2014 dataset: a) Highway sequence from the baseline category, b) Canoe from the dynamic background category, c) LakeSide from the thermal category, d) Port-17fps from the low frame rate category, and e) WetSnow from the bad weather category.

a sequence. If a scene background contains dynamics or turbulence, then these samples may cover some of these changes. However, the training set is unlikely to cover all the scene background dynamics due to noise from the sensors, and thus, the model will not learn the scene background by rote. After the training stage, we first infer all the frames in the image sequence and further compute the quantitative results by using the evaluation toolkit provided by the website of the CD2014 dataset. Thanks the lightweight design of the end-to-end model, the entire experiment on CD2014 takes less then 3 hours, including all the computational overheads of processing image data, training models, inference, output storage, and evaluations.

Table 3 provides the detailed results by using several common evaluation criteria on all the categories of challenges in the CD2014 dataset. The evaluation criteria includes Specificity, False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), Precision, Recall, and F-Measure. Among these criteria, we emphasize PWC and F-Measure because they are functions of and consider the values of other former metrics. F-Measure is the weighted harmonic mean of Precision and Recall; it is presented as follows:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall}. \tag{6}$$

PWC is a function of false negative (FN), false positive (FP), true negative (TN), and true positive (TP); it is given as

$$PWC = 100 * \frac{FN + FP}{TP + FN + FP + TN}, \tag{7}$$

where FN and FP are false negative and false positive errors, respectively. When a PWC value is less than 1, the output results in the form of mask are basically consistent with the groudtruths. All the PWC values of our methods are under 0.2, indicating that our segmented foreground masks are mostly correct through the timeline in the video. As indicated in [32], the F-measure of ordinary annotations from human beings is approximately 0.948. Most our F-measure values are over 0.95, indicating that our method has successfully addressed the challenges in this dataset. A method exhibits high potential for practical applications if its performance is higher than that of human beings.

From the statistics in Table 3, the performance on the NightVision category is relatively low but still above the acceptable level. Through visual inspection into qualitative samples, we found that most of the images in the NightVideos category are not well white-balanced, blurred, and contain dense illumination noise, which were probably recorded by early generation surveillance cameras. The images shown in Fig. 6 are samples on which the proposed method exhibits poor performance. In the highly blurred or noisy samples, as shown in the first three columns of the figure, the proposed method experiences difficulty in predicting the edges of the foreground objects. The last column contains samples from "intermittentPan" in the PTZ category. The proposed method is unable to segment the foreground objects when the camera is rotating/moving fast. Except for these particular cases, the proposed method overcomes most challenges and clearly segments the foreground objects. The qualitative results show that the overall performance is satisfactory.

Some of the qualitative results are presented in Fig. 5, which also presents a comparison with other state-of-the-art methods. The output of our method clearly outperforms the latest unsupervised methods, such as SuBSENSE [26] and SWCD [30]. The unsupervised methods experience difficulties in correctly segmenting foreground objects when the background is in a dynamic state, such as the image sequence in the Camera Jitter category,F or the foreground objects with shadows. Supervised methods, such as Cascade-CNN [32], FgSegNet-S [11], and our proposed method, can address the challenge of dynamic variance in background. Compared with that of Cascade-CNN [32], our results contain less noise, which is false positive errors. Cascade-CNN [32] is prone to be affected by shadows or color shift, although texture remains unchanged. Compared with FgSegNet-S [11], the qualitative results are mostly at the same level with few noticeable differences. However, as shown

Table 2
Comparison of F-measure of the state-of-the-art methods on CD2014 dataset.

Method	baseline	bad. weat.	cam. jitter	dyna. bg	int. obj.m	low f. rate	night vid.	PTZ	shadow	thermal	turbul.	Overall
Adaptive GMM [15]	0.433	0.3704	0.3113	0.2087	0.2626	0.4617	0.3684	0.2879	0.3445	0.2179	0.2559	0.3202
FuzzySpatialCoh [18]	0.9254	0.7762	0.6405	0.5953	0.5464	0.4613	0.4056	0.0190	0.5899	0.8715	0.4685	0.5883
BMN-BSN [40]	0.9371	0.8531	0.6069	0.5662	0.5714	0.7045	0.6113	N/A	0.528	0.8421	0.5503	0.6771
GMM-Spatial Info [16]	0.8956	0.7815	0.8365	0.8749	0.3885	0.5785	0.4372	0.5785	0.8997	0.7727	0.6943	0.7001
SuBSENSE [26]	0.9503	0.8594	0.8152	0.8177	0.6569	0.6594	0.4918	0.3894	0.8986	0.8171	0.8423	0.7453
PAWCS [27]	0.9397	0.8059	0.8137	0.8938	0.7764	0.6433	0.4171	0.445	0.8934	0.8324	0.7667	0.7477
ShareModel [49]	0.9559	0.8072	0.8034	0.8036	0.6836	0.794	0.4794	0.4569	0.8902	0.7889	0.8493	0.7557
DeepBS [33]	0.958	0.8647	0.899	0.8761	0.6097	0.59	0.6359	0.3306	0.9304	0.7583	0.8993	0.7593
IUTIS-5 [50]	0.9567	0.8289	0.8332	0.8902	0.7296	0.7911	0.5132	0.4703	0.9084	0.8303	0.8507	0.782
BSUV-Net [38]	0.9640	0.8730	0.7788	0.8176	0.7601	0.6788	0.6815	0.6562	0.9664	0.8455	0.7631	0.7986
Structured-GsMM [17]	0.95	0.86	0.82	0.85	0.82	0.75	0.51	N/A	0.89	0.85	0.85	0.82
SFEN(VGG) [12]	0.9594	0.8949	0.9422	0.7356	0.7538	0.6175	0.7526	0.7816	0.9084	0.8546	0.9207	0.8292
SFEN(ResNet50) [12]	0.9294	0.9461	0.9518	0.822	0.8453	0.808	0.8585	0.7776	0.9647	0.9444	0.8011	0.8772
Cascade [32]	0.9786	0.9451	0.9758	0.9658	0.8505	0.8804	0.8926	0.9344	0.9593	0.8958	0.9215	0.9272
FgSegNet-S [11]	0.998	0.9902	0.9951	0.9952	0.9942	0.9511	0.9837	0.988	0.9967	0.9945	0.9796	0.9878
Ours	0.9902	0.9608	0.9898	0.9907	0.9826	0.9586	0.9496	0.9739	0.9878	0.9828	0.9759	0.9766

Table 3
Detailed results using commom evaluation critera on all the categories of challenges in the CD2014 dataset.

Category	Specificity	FPR	FNR	PWC	Precision	Recall	F-Measure
PTZ	0.99974	0.00026	0.02687	0.05353	0.97468	0.97313	0.97388
shadow	0.9995	0.0005	0.01241	0.1002	0.98792	0.98759	0.98775
baseline	0.99966	0.00034	0.00953	0.0591	0.9899	0.99047	0.99018
cam.jitter	0.99953	0.00047	0.00932	0.08685	0.98897	0.99068	0.98981
dyna.bg	0.9999	0.0001	0.00939	0.02175	0.99083	0.99061	0.99071
night vid.	0.99916	0.00084	0.05636	0.18777	0.95575	0.94364	0.94958
low f.rate	0.99966	0.00034	0.04227	0.07024	0.95954	0.95773	0.95861
thermal	0.99926	0.00074	0.02004	0.17719	0.98562	0.97996	0.98277
int.obj.m	0.99938	0.00062	0.01932	0.13745	0.98463	0.98068	0.98263
turbul.	0.99986	0.00014	0.02473	0.03061	0.97664	0.97527	0.97594
bad.weat.	0.99953	0.00047	0.03929	0.09781	0.96128	0.96071	0.96077
mean	0.99956	0.00044	0.0245	0.09295	0.9778	0.9755	0.9766

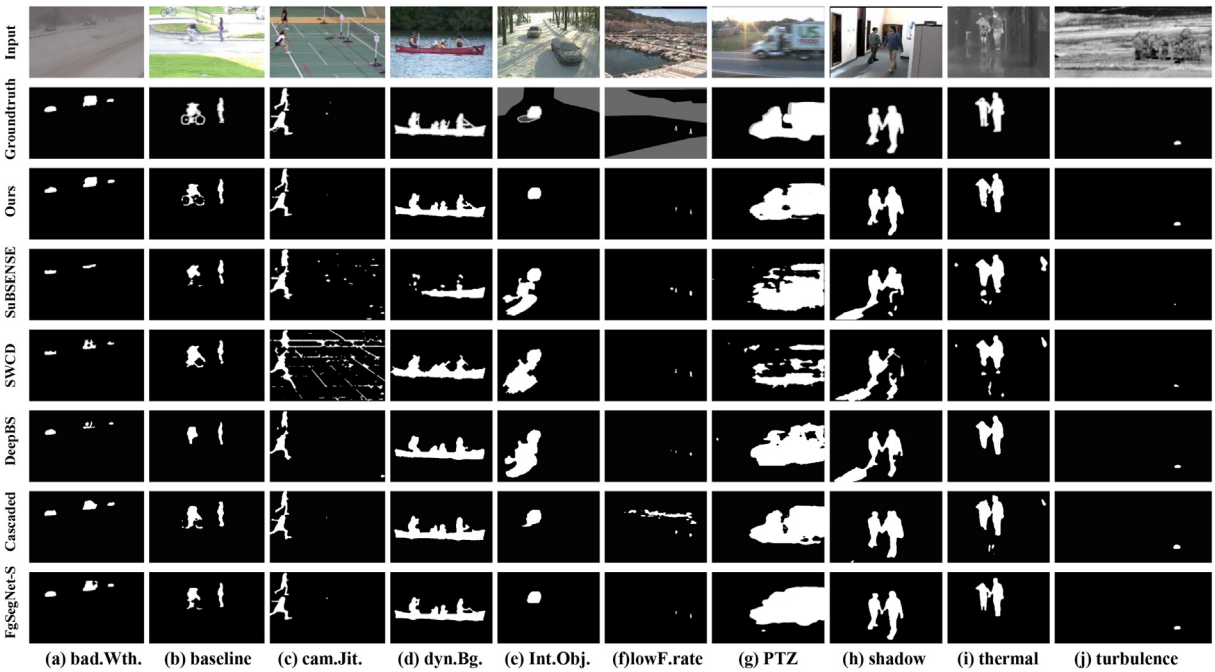
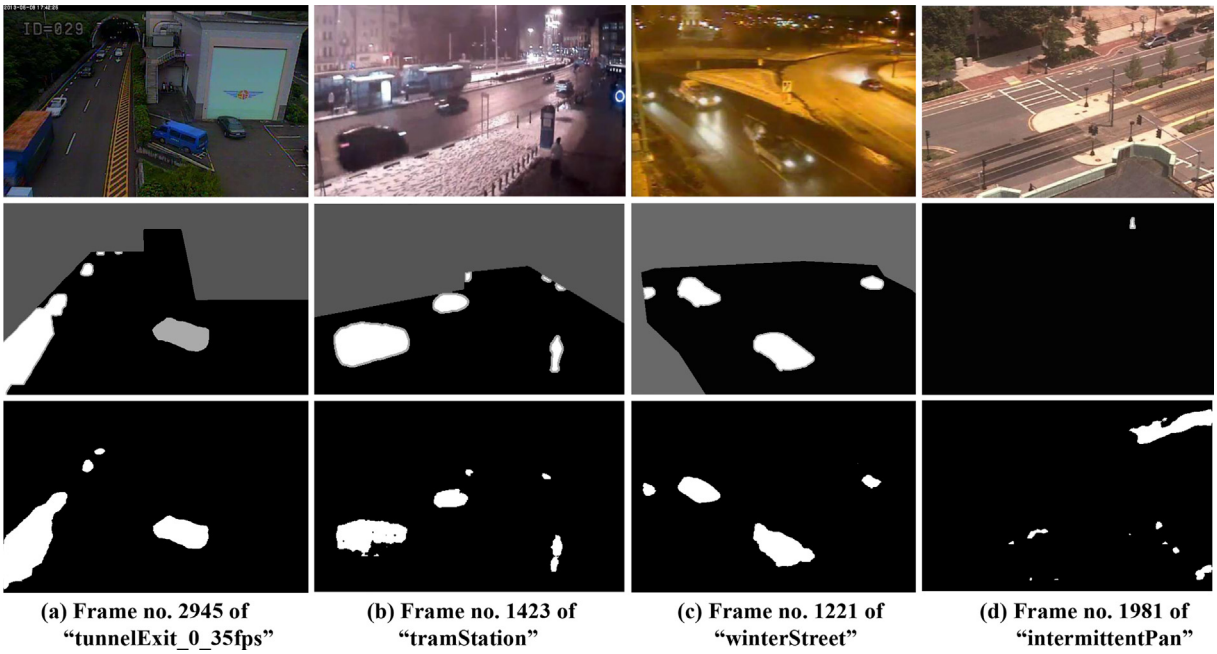


Fig. 5. Comparison of some qualitative results on the CD2014 dataset, from top to bottom: input RGB images, groundtruths of input, our results, and results of SuBSENSE [26], SWCD [30], DeepBS [33], Cascade-CNN [32], and FgSegNet-S [11].



(a) Frame no. 2945 of "tunnelExit_0_35fps" (b) Frame no. 1423 of "tramStation" (c) Frame no. 1221 of "winterStreet" (d) Frame no. 1981 of "intermittentPan"

Fig. 6. Challenging cases from the CD2014 dataset. Top row: input images. Middle row: groundtruths. Bottom row: our results.

in Fig. 5(a) and 5(b), we determine through all the result sequences that our method appears more robust to the challenge of segmenting foreground objects from a similar background scene. In general, our qualitative results are mostly consistent with groundtruth mask. However, our method performs relatively weak on the PTZ category, which is one of the most challenging categories in the dataset for existing best methods. Image sequences in the PTZ category are not from conventional surveillance cases but captured when a camera is moving or rotating. The moving and non-static background scene contains more visual information, and this information is too much for a lightweight model to encode fully, leading to relatively poor performance.

Table 2 presents the comparisons of the results in F-measure with other state-of-the-art methods on different categories of the CD2014 dataset. In the comparison, our lightweight model outperforms the other methods except for its marginal underperformance compared with FgSegNet-S [11], which is a VGG-based heavy model. Some comparative data in the table are from the CD2014 benchmark webpage.[1]

4.2. Experimental analysis of the SBI2015 dataset

For the experiments on the SBI2015 dataset, the training and validation sets are split by 20% versus 80% of entire image sequences. Given that the image sequences in this dataset are shorter than those in the CD2014 dataset, some training sets may be small and the model may not be well trained. That is, the powerful encoding capability of CNN-based methods is not fully utilized because support from data is not sufficiently big. Table 4 provides the detailed results by using several common evaluation criteria on all the image sequences in the SBI2015 dataset. Some image sequences present unconventional challenges. In the Snellen sequence, for example, foreground objects are fast moving and a large area of motion blur caused by a low capturing rate of the camera is observed. The motion blur is semi-transparent, making determining foreground area difficult, even for human beings. For the Toscana sequence, we encounter a similar problem as that in [32,11], i.e., the lack of sufficient training data. That is, only two images are available for training and the entire sequence contains only six images. Data is insufficient for training a model, and the model will probably end up underfitted. In the CaVignal sequence, the foreground object remains static most of the time in two locations, posing a challenge to some methods because several conventional methods may regard a long-time static object as background. The proposed method avoids this problem when a clearly labeled foreground sample and a sample with long-blocking background texture are provided in the training set. The detailed sequence-wise comparisons with state-of-the-art methods in the F-measure and PWC are provided in Table 5. Our overall performance is comparable with those of other CNN-based heavyweight models. In some image sequences, e.g., Board, CAVIAR1, Candela_m1.10, CAVIAR2, and CaVignal, the proposed method outperforms others. Additional qualitative test results on the SBI2015 dataset are presented in Fig. 7.

[1] The benchmark of the CD2014 dataset: http://jacarini.dinf.usherbrooke.ca/results2014.

Table 4
Detailed results of the proposed method on the SBI2015 dataset.

Sequence Name	Specificity	FPR	FNR	PBC	Precision	Recall	F-Measure
Board	0.99872	0.00128	0.00208	0.15193	0.99695	0.99792	0.99743
CAVIAR1	0.99991	0.00009	0.00119	0.01273	0.99778	0.99881	0.9983
CAVIAR2	0.99985	0.00015	0.04436	0.0321	0.96216	0.95564	0.95889
CaVignal	0.99872	0.00128	0.01757	0.31843	0.99019	0.98243	0.98629
Candela	0.99965	0.00035	0.0105	0.07565	0.99156	0.9895	0.99053
Foliage	0.93924	0.06076	0.03474	4.25685	0.97363	0.96526	0.96943
HallAndMonitor	0.9996	0.0004	0.02328	0.09516	0.98356	0.97672	0.98013
HighwayI	0.99905	0.00095	0.01138	0.19869	0.9913	0.98862	0.98996
HighwayII	0.99974	0.00026	0.00747	0.04757	0.99164	0.99253	0.99208
HumanBody2	0.99794	0.00206	0.02285	0.41771	0.98175	0.97715	0.97944
IBMtest2	0.9988	0.0012	0.04183	0.30701	0.97462	0.95817	0.96633
PeopleAndFoliage	0.99305	0.00695	0.01149	0.93303	0.99363	0.98851	0.99106
Snellen	0.95978	0.04022	0.01351	2.54169	0.96826	0.98649	0.97729
Toscana	0.98423	0.01577	0.28829	7.36909	0.92412	0.71171	0.80412
Overall	0.99059	0.00941	0.0379	1.19697	0.98008	0.9621	0.97009

Table 5
Detailed quantitative comparison in terms of F-measure and PWC with state-of-the-art methods on the SBI2015 dataset.

Image Sequence	Cascade [32]		FgSegNet-S [11]		Ours	
	PWC	F-measure	PWC	F-measure	PWC	F-measure
Board	0.3	0.99	0.1364	0.9977	0.1519	0.9974
Candela_m1.10	0.12	0.98	0.0507	0.9936	0.0127	0.9983
CAVIAR1	0.03	0.995	0.0097	0.9987	0.0321	0.9589
CAVIAR2	0.04	0.95	0.0136	0.9826	0.3184	0.9863
CaVignal	0.58	0.97	0.3158	0.9864	0.0757	0.9905
Foliage	6.31	0.95	3.8221	0.9726	4.2569	0.9694
HallAndMonitor	0.16	0.97	0.0394	0.9918	0.0952	0.9801
HighwayI	0.3	0.98	0.1457	0.9926	0.1987	0.9900
HighwayII	0.1	0.98	0.0299	0.9950	0.0476	0.9921
HumanBody2	0.77	0.96	0.1653	0.9919	0.4177	0.9794
IBMtest2	0.48	0.95	0.1372	0.9850	0.3070	0.9663
PeopleAndFoliage	1.46	0.99	0.9382	0.9910	0.9330	0.9911
Snellen	45.84	0.33	2.3244	0.9790	2.5417	0.9773
Toscana	21.63	0.51	3.8057	0.9060	7.3691	0.8041
Overall	5.58	0.8932	0.8524	0.9831	1.1970	0.9701

4.3. Ablative study

Ablative studies are conducted in two directions: 1) variations in depth and scale and 2) existence of skip connections. Testing ablated alternatives in different scales helps determine the optimal size of the scalable foreground segmentation model. Meanwhile, ablating skip connections allows us to verify the effectiveness of the skip connections. To change the capacity of the scalable model, we mostly adjust the N and ℓ parameters to set up a model. N and ℓ represent the depth of the encoder/decoder and the thickness of its layers, respectively. Table 6 provides the F-measure results on the CD2014 dataset using grid search for the optimal N and ℓ. When ℓ increases, performance rises from one level to another level. The primary reason is that, when the capacity of each layer increases, the model is considerably easier to train fully and fit all the trainig data. Performance is evaluated on the entire dataset, but some sequences are considerably difficult. Thus, if the model fails training on a challenging sequence, then the performance of F-measure on that sequence is nearly 0 and overall performance drops to a lower level. In particular, given a lower N or ℓ value, the model is unable to be trained well or sufficiently fit the data, leading to underfitting or failure to construct a discriminative model. For instance, the scalable model with $N = 3$ and $\ell = 16$ fails to train a model for the "turbulence2" and "port_0_17 fps" sequences. The foreground objects in these sequences are too small, and a model requires more capacity to distinguish noisy background information and tiny foreground objects with lower level details. Thus, increasing model capacity with a higher N and ℓ values solves the problem. From the data in Table 6, two conclusions can be drawn. 1) When depth N is less than 3, the results are less satisfactory; nevertheless, a value higher than 3 will no longer be helpful in increasing performance. That is, $N = 3$ is the optimal value. 2) ℓ, which mostly contributes to the capacity, should be over 24. When ℓ is higher than 36 and $N \geqslant 3$, the performance fluctuates to approximately 0.97. Considering that computational cost is proportional to ℓ, $\ell = 32$ is the optimal value that balances cost with performance. To verify the effectiveness of the long and short skip connections separately, the proposed scalable architecture is further tested by removing some components, i.e., long skip connections (LScn), asymmetric bottleneck (ASneck), and short skip connections (SScn). Table 7 presents the F-measure results on the CD2014 dataset from ablated architectures. On the one hand, the results show that long skip connection is helpful in handling general chal-

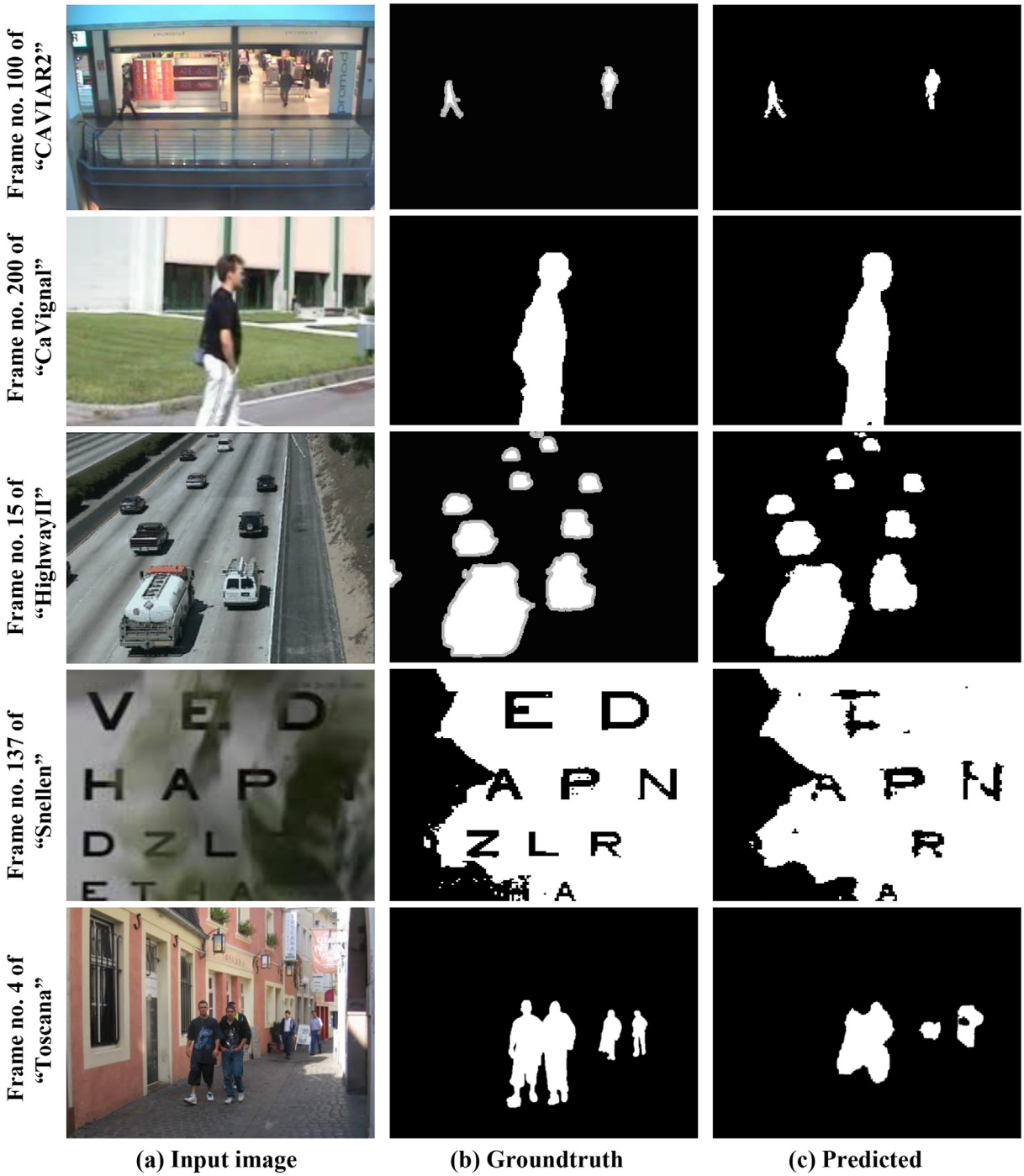


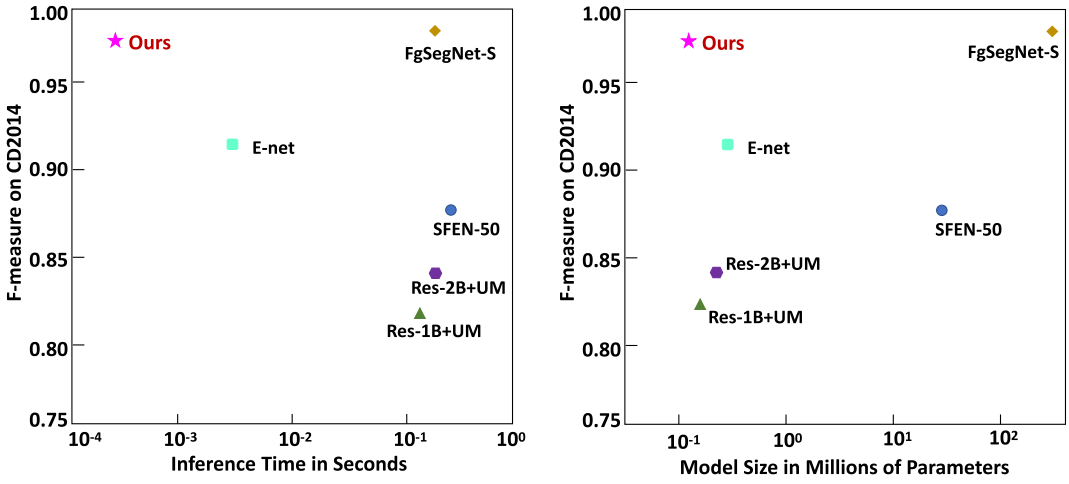
Fig. 7. Qualitative results of the SBI2015 dataset generated using the proposed method.

Table 6
F-measure results on the CD2014 dataset from scalable models with different N and ℓ values.

<*N*, *l*>	16	24	32	48	64
2	0.8326	0.8933	0.9275	0.9302	0.9287
3	0.8809	0.9479	0.9766	0.9765	0.9841
4	0.8541	0.9504	0.9739	0.9780	0.9833
5	0.8760	0.9523	0.9763	0.9780	0.9864

Table 7
F-measure results on the CD2014 dataset from the ablated architectures.

Model Parameters	Without LScn	Without ASneck	Without SScn	Full Architecture
<N = 3, l = 32>	0.8628	0.9396	0.9469	0.9766
<N = 3, l = 64>	0.8596	0.9402	0.9532	0.9841



(a) F-measure vs. inference time (b) F-measure vs. model size

Fig. 8. F-measure and cost comparisons with state-of-the-art methods. a) F-measure results on CD2014 vs. inference time in seconds. b) F-measure and model size comparison. The model sizes of FgSegNet-S and SFEN-50 are given by their lowest bound measured on their backbone networks. All the data in this figure are related to Table 8.

lenges. On the other hand, using short skip connection is effective in enhancing and fine-tuning the results, driving the F-measure to exceed 0.97.

4.4. Computational cost and compared to lightweight models

In addition to evaluating the proposed method on the two datasets and comparing the results with other state-of-the-art methods, we also conduct experiments to measure various computational costs of the proposed method and compare them with those of other potential lightweight solutions. In this subsection, we implement commonly used end-to-end lightweight architectures, including E-net [48], a model with two ResBlock [42] as encoder and an up-sampling module (UM) as decoder, and a model with one ResBlock as encoder and UM as decoder. Table 8 provides the detailed computational costs of the tested models and their corresponding performance on the two datasets, and the comparison with other methods is presented in Fig. 8. To demonstrate how lightweight the proposed method is, the table includes the state-of-the-art non-lightweight model FgSegNet-S [11] for comparison. Computational costs are measured on the basis of the following: the memory space required to store a trained model in external memory; the number of parameters in the trained model; the G-FLOPs, which is the scale of computations; the time used to train an epoch with a typical 200-image training set; and the time for the single-pass inference of data from one end to the other end of the model. Although space demand in external memory depends on the development environment, the size of a saved model is relatively consistent with the size of parameters plus a small overhead information. From the data presented in Table 8, the proposed method is shown to be extremely lightweight by these measurements: model storage only takes 435 kBytes in hard disk; and model parameters are only 105,000 float type numbers, i.e., a three-order decrease in model size from the best performing state-of-the-art model FgSegNet-S [11]. Although our method slightly underperforms in segmentation accuracy compared with the best performing method, our advantage is still significant: more than 99.9% improvement in computational cost. Compared to other lightweight end-to-end model, our method requires less computational costs and exhibits better performance. In addition, running speed in frames per second (FPS) is over 250 FPS in inference, which considerably exceeds real-time requirement.[2] Thanks the lightweight nature of the proposed method, applicable scenarios are widely extended, enabling the application of the method to smart terminal devices with limited computational capacity and low memory space. Moreover, general human

[2] The model processes the data with high efficiency. Thus, the model may no more be the bottleneck in the processing speed of this computation pipeline. However, the hard disk can be the slowest part in the pipeline. Using a solid-state disk is highly recommended when reproducing the experimental results.

Table 8
Comparison of computational costs by using various measurements.

Method	File Size (Kbytes)[a]	Num. Para.[b]	G FLOPs	Train/ Epoch[c]	Infer. Time[d]	CD2014 Fm	CD2014 PWC	SBI2015 Fm	SBI2015 PWC
SFEN(ResNet50) [12]	>97,800.0	>25.5	>3.60	–	≈5fps	0.877	–	–	–
FgSegNet-S [11]	>520,000.0	>140.00	>15.50	–	≈60fps	0.9878	–	–	–
Res-2B + UM [42]	798.9	0.197	0.97	≈2.3 s	≈60fps	0.8414	2.2999	0.4156	3.73
Res-1B + UM [42]	500.7	0.123	0.66	≈2s	≈70fps	0.8299	1.9059	0.4147	3.71
E-net [48]	1499.8	0.351	0.46	≈10 s	≈17fps	0.9135	0.2441	0.8447	2.4
Ours	432.3	0.105	0.13	≈1.43 s	>250fps	0.9766	0.093	0.9701	1.1

[a] The filesize is measured on pytorch checkpoint, i.e. '*.pth' file, in where parameters of a trained model are externally stored.
[b] In million; type of parameter is Float.
[c] Each epoch traverses 200 training samples.
[d] Testing time includes overheadings of image transformations.

level performance in F-measure is approximately 0.95 [32], indicating that the confident F-measure of the groundtruths is roughly at this level. Our method has reached human level with much higher efficiency and significantly exceeds real-time inference speed.

5. Conclusion

In this work, we proposed a novel lightweight end-to-end model for segmenting foreground objects against the scene's background. In contrast with other state-of-the-art methods that merely emphasize boosting segmentation accuracy, we focused on segmentation performance and efficiency. The proposed method is a lightweight U-like net with improved modifications: 1) long and short skip connections are used among convolutional blocks and within the bottleneck block, and 2) element-wise summing-based feature fusion is used before each up-sampling layer. Experiments were performed on the popular CD2014 and SBI2015 datasets for the foreground segmentation problem. The experiments showed that the proposed method reached human-level performance and achieved breakthroughs in efficiency in terms of training and inference. The efficiency of inference considerably exceeded real-time level with 250+ FPS. Apart from maintaining high segmentation accuracy, the proposed model requires an extremely low computational cost of approximately 0.105 million parameters and only 0.13 G-FLOPs, which is less than 0.001 times compared with current state-of-the-art methods and generates more than 99.9% improvement in terms of memory and computational costs. The experimental results were compared with those of other state-of-the-art methods qualitatively and quantitatively.

On the basis of this work, the further research can be potentially conducted in two aspects. In engineering, the proposed method can be implemented in smart terminal devices for applications in motion detection and can be integrated into a larger IoT system. Thanks the lightweight nature of the proposed method, the model can potentially inference foreground objects in smart terminals. A server with higher computational capacity can then manage these terminal devices by initializing the model or pushing latest updates. The further research is likely to be conducted in the domain of edge computing. In a theoretical study, given its efficiency and low computational cost, the use of the proposed method as a component in a solution for higher-level problems, e.g., as an attention generator in anomaly detection for intelligent video surveillance, is worth investigating. Through foreground object masks, an object detector can focus on recognizing foreground objects and tracking their traces. These extensions could can function key preprocessing components in an anomaly detection system.

CRediT authorship contribution statement

Cong Lin: Conceptualization, Methodology, Software, Data curation, Writing - original draft. **Shijie Zhang:** Methodology, Software. **Shaodi You:** Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Xiaoxiang Liu:** Conceptualization, Writing - review & editing. **Zhiyu Zhu:** Conceptualization, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities of China (Grant No. 11618316), the Science & Technology Development Fund of Tianjin Education Commission for Higher Education (Grant No. 2018KJ082), and the National Natural Science Foundation of China (Grant No. 62006101).

References

[1] R.J. Radke, S. Andra, O. Al-Kofahi, B. Roysam, Image change detection algorithms: a systematic survey, IEEE Trans. Image Process. 14 (3) (2005) 294–307, https://doi.org/10.1109/TIP.2004.838698.

[2] Z. Chen, R. Wang, Z. Zhang, H. Wang, L. Xu, Background-foreground interaction for moving object detection in dynamic scenes, Inf. Sci. 483 (2019) 65–81, https://doi.org/10.1016/j.ins.2018.12.047, http://www.sciencedirect.com/science/article/pii/S0020025518309903.

[3] X. Wang, M. Wang, W. Li, Scene-specific pedestrian detection for static video surveillance, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 361–374, https://doi.org/10.1109/TPAMI.2013.124.

[4] C.-M. Pun, C. Lin, A real-time detector for parked vehicles based on hybrid background modeling, J. Vis. Commun. Image Represent. 39 (2016) 82–92, https://doi.org/10.1016/j.jvcir.2016.05.009, http://www.sciencedirect.com/science/article/pii/S1047320316300761.

[5] Y. Xian, X. Rong, X. Yang, Y. Tian, Evaluation of low-level features for real-world surveillance event detection, IEEE Trans. Circuits Syst. Video Technol. 27 (3) (2017) 624–634, https://doi.org/10.1109/TCSVT.2016.2589838.

[6] C. Stauffer, W.E.L. Grimson, Adaptive background mixture models for real-time tracking, in: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Vol. 2, 1999, pp. 246–252 Vol. 2. https://doi.org/10.1109/CVPR.1999.784637..

[7] K. Them, M.G. Kaul, C. Jung, M. Hofmann, T. Mummert, F. Werner, T. Knopp, Sensitivity enhancement in magnetic particle imaging by background subtraction, IEEE Trans. Med. Imaging 35 (3) (2016) 893–900, https://doi.org/10.1109/TMI.2015.2501462.

[8] A. Basharat, A. Gritai, M. Shah, Learning object motion patterns for anomaly detection and improved object detection, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8. https://doi.org/10.1109/CVPR.2008.4587510..

[9] O. Barnich, M.V. Droogenbroeck, Vibe: A universal background subtraction algorithm for video sequences, IEEE Trans. Image Process. 20 (6) (2011) 1709–1724, https://doi.org/10.1109/TIP.2010.2101613.

[10] A. Cuzzocrea, E. Mumolo, A novel gpu-aware histogram-based algorithm for supporting moving object segmentation in big-data-based iot application scenarios, Inf. Sci. 496 (2019) 592–612, https://doi.org/10.1016/j.ins.2019.03.029.

[11] L.A. Lim, H.Y. Keles, Foreground segmentation using convolutional neural networks for multiscale feature encoding, Pattern Recogn. Lett. 112 (2018) 256–262, https://doi.org/10.1016/j.patrec.2018.08.002.

[12] Y. Chen, J. Wang, B. Zhu, M. Tang, H. Lu, Pixelwise deep sequence learning for moving object detection, IEEE Trans. Circuits Syst. Video Technol. 29 (9) (2019) 2567–2579, https://doi.org/10.1109/TCSVT.2017.2770319.

[13] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, P. Ishwar, Cdnet 2014: An expanded change detection benchmark dataset, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 393–400. https://doi.org/10.1109/CVPRW.2014.126..

[14] L. Maddalena, A. Petrosino, Towards benchmarking scene background initialization, in: V. Murino, E. Puppo, D. Sona, M. Cristani, C. Sansone (Eds.), New Trends in Image Analysis and Processing – ICIAP 2015 Workshops, Springer International Publishing, Cham, 2015, pp. 469–476.

[15] Z. Zivkovic, Improved adaptive gaussian mixture model for background subtraction, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004, Vol. 2, 2004, pp. 28–31 Vol. 2. https://doi.org/10.1109/ICPR.2004.1333992..

[16] A. Boulmerka, M.S. Allili, Foreground segmentation in videos combining general gaussian mixture modeling and spatial information, IEEE Trans. Circuits Syst. Video Technol. 28 (6) (2018) 1330–1345, 10.1109TCSVT.2017.2665970.

[17] G. Shi, T. Huang, W. Dong, J. Wu, X. Xie, Robust foreground estimation via structured gaussian scale mixture modeling, IEEE Trans. Image Process. 27 (10) (2018) 4810–4824, https://doi.org/10.1109/TIP.2018.2845123.

[18] L. Maddalena, A. Petrosino, A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection, Neural Comput. Appl. 19 (2) (2010) 179–186.

[19] E. Ontiveros, J. Gonzalez, Design and fpga implementation of real-time edge detectors based on interval type-2 fuzzy systems, J. Multiple-Valued Logic Soft Comput. 33 (2019) 295.

[20] C.I. Gonzalez, P. Melin, J.R. Castro, O. Castillo, Edge detection approach based on type-2 fuzzy images, J. Multiple-Valued Logic Soft Comput. 33 (2019) 431–458.

[21] P. Melin, C.I. Gonzalez, J.R. Castro, O. Mendoza, O. Castillo, Edge-detection method for image processing based on generalized type-2 fuzzy logic, IEEE Trans. Fuzzy Syst. 22 (6) (2014) 1515–1525.

[22] G.E. Martínez, C.I. Gonzalez, O. Mendoza, P. Melin, General type-2 fuzzy sugeno integral for edge detection, J. Imaging 5 (8) (2019), https://doi.org/10.3390/jimaging5080071.

[23] C.I. Gonzalez, P. Melin, J.R. Castro, O. Castillo, O. Mendoza, Optimization of interval type-2 fuzzy systems for image edge detection, Appl. Soft Comput. 47 (2016) 631–643, https://doi.org/10.1016/j.asoc.2014.12.010.

[24] M. Hofmann, P. Tiefenbacher, G. Rigoll, Background segmentation with feedback: The pixel-based adaptive segmenter, in: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2012, pp. 38–43. https://doi.org/10.1109/CVPRW.2012.6238925..

[25] R. Zhang, W. Gong, V. Grzeda, A. Yaworski, M. Greenspan, Scene dynamics estimation for parameter adjustment of gaussian mixture models, IEEE Signal Process. Lett. 21 (9) (2014) 1130–1134, https://doi.org/10.1109/LSP.2014.2326916.

[26] P. St-Charles, G. Bilodeau, R. Bergevin, Subsense: A universal change detection method with local adaptive sensitivity, IEEE Trans. Image Process. 24 (1) (2015) 359–373, https://doi.org/10.1109/TIP.2014.2378053.

[27] P. St-Charles, G. Bilodeau, R. Bergevin, A self-adjusting approach to change detection based on background word consensus, in: 2015 IEEE Winter Conference on Applications of Computer Vision, 2015, pp. 990–997. https://doi.org/10.1109/WACV.2015.137..

[28] S. Jiang, X. Lu, Wesambe: A weight-sample-based method for background subtraction, IEEE Trans. Circuits Syst. Video Technol. 28 (9) (2018) 2105–2115, https://doi.org/10.1109/TCSVT.2017.2711659.

[29] A. Zheng, T. Zou, Y. Zhao, B. Jiang, J. Tang, C. Li, Background subtraction with multi-scale structured low-rank and sparse factorization, Neurocomputing (2018), https://doi.org/10.1016/j.neucom.2018.02.101, http://www.sciencedirect.com/science/article/pii/S0925231218309494.

[30] S. Isik, K. Özkan, S. Günal, Ömer Nezih Gerek, Swcd: a sliding window and self-regulated learning-based background updating method for change detection in videos, J. Electron. Imaging 27 (2) (2018) 1–1111, https://doi.org/10.1117/1.JEI.27.2.023002.

[31] K. Lim, W. Jang, C. Kim, Background subtraction using encoder-decoder structured convolutional neural network, in: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2017, pp. 1–6. https://doi.org/10.1109/AVSS.2017.8078547..

[32] Y. Wang, Z. Luo, P.-M. Jodoin, Interactive deep learning method for segmenting moving objects, Pattern Recognition Letters 96 (2017) 66–75, scene Background Modeling and Initialization. https://doi.org/10.1016/j.patrec.2016.09.014. URL http://www.sciencedirect.com/science/article/pii/S0167865516302471..

[33] M. Babaee, D.T. Dinh, G. Rigoll, A deep convolutional neural network for video sequence background subtraction, Pattern Recogn. 76 (2018) 635–649, https://doi.org/10.1016/j.patcog.2017.09.040, http://www.sciencedirect.com/science/article/pii/S0031320317303928.

[34] R. Wang, F. Bunyak, G. Seetharaman, K. Palaniappan, Static and moving object detection using flux tensor with split gaussian models, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 420–424. https://doi.org/10.1109/CVPRW.2014.68..

[35] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.2016.2644615.

[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Computer Science, 2014.

[37] D. Zeng, M. Zhu, Background subtraction using multiscale fully convolutional network, IEEE Access 6 (2018) 16010–16021, https://doi.org/10.1109/ACCESS.2018.2817129.

[38] M. Ozan Tezcan, P. Ishwar, J. Konrad, BSUV-Net: A Fully-Convolutional Neural Network for Background Subtraction of Unseen Videos, arXiv e-prints (2019) arXiv:1907.11371..

[39] O. Ronneberger, P.Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI), Vol. 9351 of LNCS, Springer, 2015, pp. 234–241, (available on arXiv:1505.04597 [cs.CV]). URL http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a..
[40] V.M. Mondéjar-Guerra, J. Rouco, J. Novo, M. Ortega, An end-to-end deep learning approach for simultaneous background modeling and subtraction, British Machine Vision Conference (BMCV) (2019) 266.
[41] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 834–848, https://doi.org/10.1109/TPAMI.2017.2699184.
[42] K. He, X. Zhang, S. Ren, J. Sun, in: Deep residual learning for image recognition, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society, Los Alamitos, CA, USA, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.
[43] W. Sun, S. You, J. Walker, K. Li, N. Barnes, Structural edge detection: A dataset and benchmark, in: 2018 Digital Image Computing: Techniques and Applications (DICTA), 2018, pp. 1–8. https://doi.org/10.1109/DICTA.2018.8615801..
[44] D. Feng, N. Barnes, S. You, C. McCarthy, Local background enclosure for rgb-d salient object detection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2343–2350. https://doi.org/10.1109/CVPR.2016.257..
[45] X. Wang, H. Ma, X. Chen, S. You, Edge preserving and multi-scale contextual neural network for salient object detection, IEEE Trans. Image Process. 27 (1) (2018) 121–134, https://doi.org/10.1109/TIP.2017.2756825.
[46] J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 694–711.
[47] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, C. Pal, The importance of skip connections in biomedical image segmentation, in: Deep Learning and Data Labeling for Medical Applications, Springer International Publishing, Cham, 2016, pp. 179–187.
[48] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, Enet: A deep neural network architecture for real-time semantic segmentation, CoRR abs/1606.02147 (2016). arXiv:1606.02147. .
[49] Yingying Chen, Jinqiao Wang, Hanqing Lu, Learning sharable models for robust background subtraction, in: 2015 IEEE International Conference on Multimedia and Expo (ICME), 2015, pp. 1–6. https://doi.org/10.1109/ICME.2015.7177419..
[50] S. Bianco, G. Ciocca, R. Schettini, Combination of video change detection algorithms by genetic programming, IEEE Trans. Evol. Comput. 21 (6) (2017) 914–928, https://doi.org/10.1109/TEVC.2017.2694160.