IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 8, 2019, accepted May 23, 2019, date of publication May 29, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919802

X-Net: A Binocular Summation Network for Foreground Segmentation

JIN ZHANG[1,2], YANG LI[1], FEIQIONG CHEN[1], ZHISONG PAN[1], XINGYU ZHOU[3], YUDONG LI[2], AND SHANSHAN JIAO[1]

[1]Command and Control Engineering College, Army Engineering University of PLA, Nanjing 210000, China
[2]Army Military Transportation University of PLA, Zhenjiang Campus, Zhenjiang 212000, China
[3]Communication Engineering College, Army Engineering University of PLA, Nanjing 210000, China

Corresponding author: Zhisong Pan (panzs@nuaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61806220 and in part by the National Key Research Development Program of China under Grant 2017YFB0802800.

ABSTRACT In foreground segmentation, it is challenging to construct an effective *background model* to learn the spatial-temporal representation of the background. Recently, deep learning-based background models (DBMs) with the capability of extracting high-level features have shown remarkable performance. However, the existing state-of-the-art DBMs deal with video segmentation as single-image segmentation and ignore temporal cues in video sequences. To exploit temporal data sufficiently, this paper proposes a multi-input multi-output (MIMO) DBM framework for the first time, which is partially inspired by the binocular summation effect in human eyes. Our framework is an X-shaped network which allows the DBM to track temporal changes in a video sequence. Moreover, each output branch of our model could receive visual signals from two similar input frames simultaneously like the binocular summation mechanism. In addition, our model can be trained end-to-end using only a few training examples without any post-processing. We evaluate our method on the largest dataset for change detection (CDnet 2014) and achieve the state-of-the-art performance by an average overall F-Measure of 0.9920.

INDEX TERMS Foreground segmentation, background subtraction, deep learning, focal loss, binocular summation.

I. INTRODUCTION

Foreground segmentation, also known as background subtraction, is a crucial task for video processing. It is fundamental for many advanced applications such as traffic monitoring [1], anomaly detection [2] and behavior recognition [3]. Given one scenario $\boldsymbol{S}$, foreground segmentation algorithms are generally achieved by building a representation of $\boldsymbol{S}$, called *Background Model* (BM), and then detecting the changing regions (*foreground*) of each incoming frame by this model [43]. Over the years, various methods have been proposed to construct a proper BM.

Statistically modeling background is a popular strategy to segment foreground objects for its efficiency. Some typical algorithms such as GMM [37], KDE [38] and PBAS [39] assume the independence among pixels and model the variation of each pixel over time. Another prevalent strategy such as RPCA [33], [41] and RNMF [32] uses the idea of dimension reduction to achieve robustness. However, these conventional approaches lack the ability to extract high-level features to represent each pixel for semantic prediction [42]. They cannot address numerous challenges such as dynamic background, illumination changes, heavy shadows, camouflage, and camera motion simultaneously.

Recently, Convolutional Neural Networks (CNNs) have proved to be a powerful extractor in learning useful feature representations from data [24], [25]. Particularly, Fully Convolutional Networks (FCNs) [19] based on transfer learning have shown competitive performance in pixel-level classification tasks [16]–[18]. Deep learning based Background Models (DBMs) have thus emerged in the spotlight and outperformed conventional methods by large margins. In general, existing DBMs can be classified into the following two categories: patch-wise and image-wise. Patch-wise approaches [7]–[10] feed the image patches to CNNs to predict foreground probabilities of the center pixels of the patches. These models are simple and small, but there are considerable overlaps between neighboring pixels, which will

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

lead to computational inefficiency and overfitting. Furthermore, fixed patch sizes may cause loss of higher contextual information, especially when objects are significantly larger than the patch. Image-wise methods address these problems by using whole resolution images to predict foreground masks. Some image-wise methods [11]–[13] deal with video sequence segmentation as single-image segmentation. These works ignore temporal coherence in videos but produce remarkable results. Other methods [14], [15] use one *target frame* and its *references* (its previous frames) to produce one foreground mask at a time. Although these DBMs take advantage of temporal data, their accuracy still fails to outperform some single-image methods, such as FgSegNet_v2 [11].

Nevertheless, we reasonably believe that we can further improve segmentation accuracy by effectively exploring temporal data. The key is to utilize the correlation between consecutive frames. Therefore, we propose a novel DBM with multi-input multi-output (MIMO) structure because of the following consideration:

1. The multi-input (MI) structure provides *references* to the *target frame*, allowing the model to see more details and discriminative features due to information processing from multiple sources at the same time.

2. The multi-output (MO) structure makes each input frame act as both a *target frame* and a *reference*, enabling the model to capture the similarities and differences among multiple frames more straightforwardly.

3. Moreover, the MO structure naturally leads to multi-output loss function and multi-task learning, which usually brings improvement on performance in inter-related tasks [13].

4. The MIMO structure might bring important prior knowledge to foreground segmentation, since the similar regions between multiple input frames generally belong to background, while foreground objects are usually in regions with differences. Therefore, the model can learn the spatial-temporal representation of the background more effectively.

However, the MIMO model suffers from several limitations when dealing with multiple frames, e.g. M frame. First, the inference of the first frame needs to wait until the M-th frame appears, which will lead to latency. Second, the larger the M, the more complex the model might be, which tends to cause overfitting under few training examples. Thus, we consider $M = 2$ as a compromise.

Furthermore, we notice the human binocular vision system which has a fundamental X-shaped structure (Fig. 1) [20]. It allows each half of the brain to receive visual signals from both eyes to generate binocular summation after fusing and superimposing monocular visual signals. The binocular summation is not a simple summation of monocular information [48]. It could lower visual thresholds (contrast thresholds), enhance the visual sensitivity and improve the function of object detection and recognition [21], [45], [47]. Moreover, these advantages are more obvious under low luminance conditions [46].

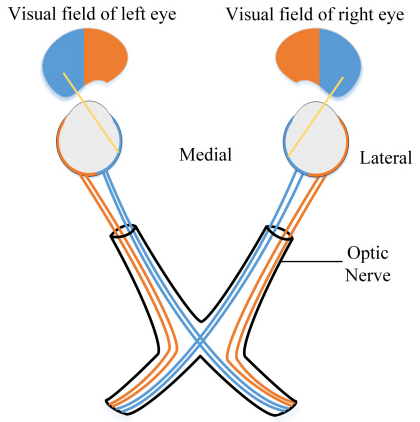


FIGURE 1. The X-shaped optic nerve in the human binocular vision system. At the chiasm, fibers from the nasal (medial, blue) half of each retina cross over to the contralateral optic tract, while fibers from the lateral (orange) halves remain ipsilateral. This X structure allows each half of the brain to receive visual signals from both eyes to generate binocular summation.

Motivated by this, we instantiate our MIMO structure into an X-shaped network, named X-Net. More specifically, it combines two branches of encoder and decoder networks via a fusion network to form an X-shaped architecture (Fig. 2). This architecture not only simulates the structure but also the mechanism of binocular summation. It perceives two similar images synchronically, extract features from them and then fuse the information. Moreover, each branch of the decoder network in the X-Net can receive visual information from both input images like the human binocular vision system.

The main contributions of this paper are as follows:

1. We propose an MIMO DBM framework for the first time, which is partially inspired by the binocular summation mechanism in human eyes. Our method effectively incorporates temporal data to learn the spatio-temporal representation of background across various scenarios.

2. We develop a novel loss function termed *soft focal loss* to address class imbalance in foreground segmentation, which is modified from the *focus loss* [30] but achieves a higher performance.

3. Our method surpasses the accuracy of all existing state-of-the-art methods on the CDnet2014 dataset.

The rest of this paper is organized as follows: Section II introduces related works of DBMs in recent years. In Section III, we describe the proposed X-Net and the soft focal loss. Section IV talks about the details of the experiment results. Section V further verifies the effectiveness of the MO structure and the soft focal loss. Finally, the conclusion of this paper is drawn in Section VI.

II. RELATED WORKS

In the last two decades, numerous BMs have been proposed, and a thorough review is beyond the scope of this paper. For an overview of non-deep-learning methods, readers can

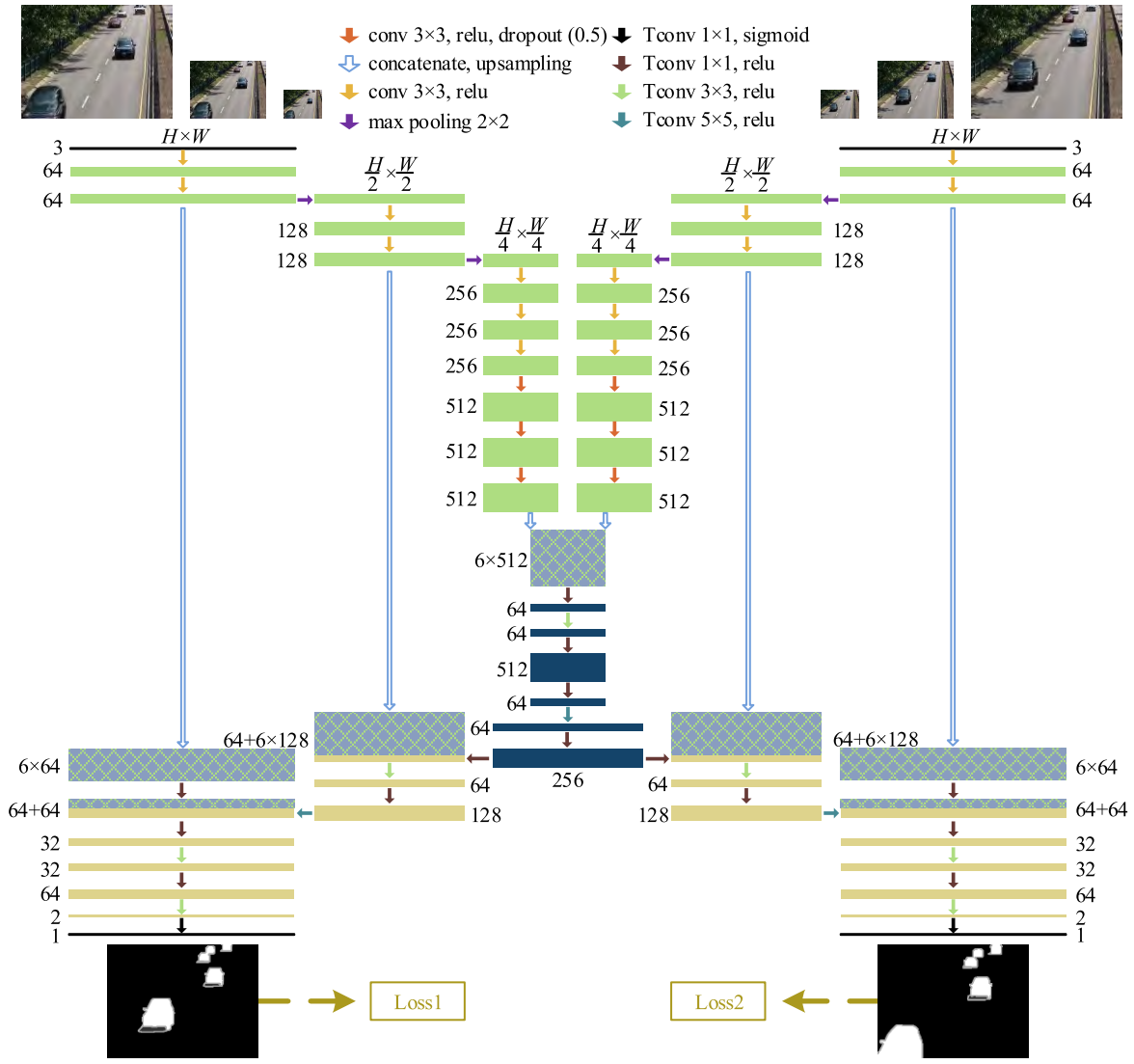


FIGURE 2. An instance of the X-Net Architecture: an encoder network (marked in green), a fusion network (marked in blue) and a decoder network (marked in yellow). Conv and Tconv represent convolution and transposed convolution operation. Each rectangle represents a feature map, with a number or an arithmetic expression to indicate the number of channels. *H* and *W* is the full resolution frame's height and width. The blue hollow arrows represent the concatenation of three-scale feature maps from two branches of the encoder after 1×, 2× and 4× up-sampling respectively. It generates the multi-scale feature maps represented by the grid rectangles in Figure.

refer to [40]. In this section, we mainly present recent DBMs from three perspectives: network architecture, temporal data usage, and loss function, which play decisive roles in model performance.

A. DBM ARCHITECTURES

Early DBMs tend to adopt CNN architectures which typically use fully connected layers after a series of convolutional layers. To avoid huge computation cost incurred by fully connected layers, Braham and Droogenbroeck [7] used image patches for training. Later on, multiple modifications were made for its insufficiency described earlier. An advanced method proposed by [9] extended basic CNN to a multi-scale CNN and trained it with multi-scaled images (including down-scaled images) to cover large objects. Latest studies such as [8] improved computational efficiency by matching the dimension of the last fully connected layers to the number of patch pixels. As an alternative, some DBMs are designed to be an encoder-decoder architecture based on FCN and perform prediction at the whole image level [16]. For example, Zhao *et al.* [13] proposed a two-stage DBM in which an encoder-decoder sub-network was employed to generate a background. Then they fed this background and a target frame to a multi-channel FCN sub-network to segment foreground objects. Conversely, without reconstructing the background, FgSegNet_M [12] proposed a triplet of encoders to extract multi-scale features and then used transposed convolution in the decoder to learn a mapping from feature space to image space. FgSegNet_S [12] adopted a single encoder but achieved high performance by applying several

parallel dilated convolutions [28] with different dilation rates to extract multi-scale features. Recently, FgSegNet_v2 [11] adapted from FgSegNet_S reached state-of-the-art results by integrating attention mechanism to learn multi-scale features without incorporating temporal data.

B. TEMPORAL DATA FOR DBMS

Proper use of temporal data might help increase segmentation accuracy while reducing DBMs' execution time. In foreground segmentation tasks, temporal data can be incorporated into DBMs by 3D or 2D convolution on multiple consecutive frames. 3D convolution, as a spatio-temporal filter, can capture motion information in video sequences [23]. Relevant studies such as [15] applied 3D convolution to track temporal changes in scenes. This model used 4 groups of chronologically parallel convolutions to process 10 consecutive input frames, then slowly merged feature maps to produce only one foreground mask for a target frame. Its multi-input single-output structure brought heavy computation burden. A more lightweight method proposed by [14] adapted VGG-16 Net [36] as the encoder and took a concatenation of 3 grayscale images (previous frame, target frame, and generated background image) as input to detect temporal changes. Although this model generated a lower computational load, its performance could be affected by information loss due to the compression of RGB images into grayscale versions. Instead of resorting to the concatenation strategy in [13] and [16], our X-Net pursues a thorough separation by feeding each branch of the encoder with one of the consecutive frames to extract complete temporal features via 2D convolution. Furthermore, multi-task learning led by MO structure usually brings improvement on performance in inter-related tasks [13]. In addition, we also notice various methods that utilize temporal coherence in video sequences to shorten inference time. For example, Shelhamer *et al.* [24] used semantic changes to motivate the updating of feature maps with different layer depths. Instead of reducing inference, our model could boost segmentation speed by parallelizing multiple encoder and decoder branches on multiple GPUs.

C. LOSS FUNCTIONS FOR DBMS

DBMs are prone to suffer from class imbalance, especially in complex scenarios in which the background pixels exceed the foreground pixels extremely [12]. In this case, the standard Cross Entropy (CE) loss function could easily be influenced by the foreground-background class imbalance. A series of variants based on the CE loss have been developed to alleviate this problem. A common method was to weight the CE loss by a class-balance factor which was set by inverse class frequency or treated as a hyper-parameter [12], [29]. These methods enforced the learning of minority (foreground) and focused on the *imbalanced quantity* (foreground/background). On the other hand, the Focal loss (FL) proposed by [30] paid attention to the *imbalanced complexity* (easy/hard). By adding a *modulating factor* to the CE loss, it down-weighted the loss assigned to well-classified (easy) examples to prevent the vast number of easy examples from overwhelming the detector. Recently, the weighted focus loss proposed by [31] integrated the above two strategies to make the model sensitive to both hard examples and minority examples. In this paper, we extend the concept of *complexity* to ***relative*** *complexity* by introducing a ***relative*** *modulating factor*. Our work can provide more competitive results compared with the focal loss, more details will be studied in discussion Section.

To sum up, existing DBMs adopt single-input single-output (SISO) structure [11]–[13], [16] or multi-input single-output (MISO) structure [15], [18] to incorporate temporal data, while the X-Net is an MIMO DBM partially inspired by binocular summation. Its two branches of the decoder network form the MO structure, which lets the model straightforwardly capture the similarities and differences between both input frames to learn the spatial-temporal property of the background effectively. Note that our method extracts multi-scale features by multi-scale input technique as FgSegNet_M, but the use of temporal data makes it even outperform FgSegNet_v2 and achieve top results on CDnet 2014 dataset (Tab. 3).

III. METHOD

A. THE X-NET ARCHITECTURE

As a simulation of the human binocular summation, our conceptual idea leads to flexible and effective designs for the X-Net to incorporate temporal data. An instance of the X-Net architecture is illustrated in Fig. 2. It includes an encoder network with two branches (marked in green), a fusion network (marked in blue), and a decoder network with two branches (marked in yellow). We also describe the exact configurations of the X-Net in blocks (Tab. 1). Such architecture allows the model to produce two foreground masks at a time.

1) THE ENCODER NETWORK

Its two branches form MI structure and work like the binocular optic nerves in human eyes, performing feature extraction from two similar frames. What's more, this MI structure provides *references* for the *target frame* and is thus favorable for extracting discriminative features via comparison. They gradually reduce the size of the feature maps and increase the number of feature channels to learn advanced and non-local features. In addition, the encoder network is designed to be a siamese [27] because of the following consideration. First, sharing the same weights means that they use the same approach to extract features from two images [26]. As the two input images are adjacent frames with similar characteristics and temporal continuity, it is natural to extract features in the same way. Second, weight sharing mechanism can reduce network parameters by half to avoid overfitting due to few training examples (usually 200 frames or less). Our encoder network can be instantiated with different backbones, but for a fair comparison with state-of-the-art models, we follow

TABLE 1. The X-Net configurations. The block 1 refers to the encoder modified from VGG-16 Net. The block 2 and 3 represent the configurations of the fusion network. The block 4 to block 6 represent the configurations of the decoder. *F* and *S* represent filter and stride respectively.

1	$\frac{H}{4} \times \frac{W}{4} \times (6 \times 512)$, VGG-16	6	$H \times W \times 2, F = 1 \times 1, S = 1$ $H \times W \times 1, F = 1 \times 1, S = 1$
2	$\frac{H}{4} \times \frac{W}{4} \times 64, F = 1 \times 1, S = 1$ $\frac{H}{4} \times \frac{W}{4} \times 64, F = 3 \times 3, S = 1$ $\frac{H}{4} \times \frac{W}{4} \times 512, F = 1 \times 1, S = 1$	5	$H \times W \times 64, F = 5 \times 5, S = 2$ $H \times W \times (64 + 64)$ $H \times W \times 32, F = 1 \times 1, S = 1$ $H \times W \times 32, F = 3 \times 3, S = 1$ $H \times W \times 64, F = 1 \times 1, S = 1$
3	$\frac{H}{4} \times \frac{W}{4} \times 64, F = 1 \times 1, S = 1$ $\frac{H}{2} \times \frac{W}{2} \times 64, F = 5 \times 5, S = 2$ $\frac{H}{2} \times \frac{W}{2} \times 256, F = 1 \times 1, S = 1$	4	$\frac{H}{2} \times \frac{W}{2} \times 64, F = 1 \times 1, S = 1$ $\frac{H}{2} \times \frac{W}{2} \times (64 + 6 \times 128)$ $\frac{H}{2} \times \frac{W}{2} \times 64, F = 3 \times 3, S = 1$ $\frac{H}{2} \times \frac{W}{2} \times 128, F = 1 \times 1, S = 1$

the practice in FgSegNet_v2 [11]. The first four blocks of VGG-16 Net [36] are adapted as the backbone of our encoder network (green path in Fig. 2).

In addition, we cannot use too many pooling layers to increase the receptive field size for extracting high-level information. This poses serious challenges to up-sample the segmentation output back to full resolution [25]. We alleviate this contradictory in a naive way, i.e., multi-scale input, as FgSegNet_M [12]. More precisely, given a pair of input images I_{L0}/I_{R0} represented in RBG color space, they are downscaled into two different scales I_{L1}/I_{R1} and I_{L2}/I_{R2}. In this paper, we use 0.5 and 0.25 for the two scales. These three pairs of images are fed simultaneously to the encoder network in parallel. This produces three pairs of outputs at three different scales: O_{L0}/O_{R0}, O_{L1}/O_{R1}, and O_{L2}/O_{R2}. After that, O_{L1}/O_{R1} and O_{L2}/O_{R2} are upscaled to match the scale of O_{L0}/O_{R0}. Finally, they are concatenated and fed to the fusion network.

In contrast to the encoder networks, the fusion network and decoder network should gradually reduce the feature channels and step-wise upscale the feature maps to the full resolution to produce foreground masks. To avoid the information loss caused by sharp decrease of the feature maps, the output feature maps of each block are designed to be step-wise reduced by half (block 2 to block 5 in Tab.1). Moreover, for computational efficiency, each block first applies 1×1 transposed convolution to project the high dimensional feature maps into 64 feature maps, then operates with the bigger kernel transposed convolution to increase the receptive field size. Note that ReLU non-linearities are applied to every layer of the X-Net, except the last layer where a sigmoid activation is used.

2) THE FUSION NETWORK

The representations learned from two branches of the encoder are merged by the fusion network (blue path in Fig. 2). Hence, each branch of the decoder network can be aware of the information from each input frame, similar to the function of the human binocular summation. To be concrete, the configurations of the fusion network is illustrated in block 2 and block 3 in Tab. 1. Due to the concatenation of features across three different scales in both branches of the encoder, the feature maps extracted by the encoder network have a large depth, i.e. $3072 = 6 \times 512$ (block 0 in Tab. 1). Therefore, the block 2 first projects this high dimensional feature maps into $\frac{H}{4} \times \frac{W}{4} \times 64$ via 1×1 transposed convolution, then use 3×3 transposed convolution for feature fusion. After that, to increase the non-linear representation ability, 1×1 transposed convolution is used again to enlarge the number of feature maps to 512. A similar process is operated in blocks 3, except that we apply 5×5 transposed convolution with a stride of 2 to upscale feature maps. In addition, the output channels of the block 3 are reduced to 256.

3) THE DECODER NETWORK

This network is MO structure which includes two independent branches (yellow path in Fig. 2) with the same configurations (from block 4 to 6 in Tab. 1). It performs detection, location, and classification at the same time. Meanwhile, to compensate for the low resolution of high-level features during up-sampling [16], feature maps from the middle and early layers are exploited by skip-connections (blue hollow arrows between the encoder and decoder in Fig. 2). More specifically, the three-scale feature maps from two branches of the encoder, after $1\times$, $2\times$ and $4\times$ up-sampling, are combined to both branches of the decoder and generate the concatenated feature maps of $\frac{H}{2} \times \frac{W}{2} \times (64 + 6 \times 128)$ in block 4 and $H \times W \times (64 + 64)$ in block 5. In block 6, the feature maps of $H \times W \times 64$ are projected into $H \times W \times 2$ using a 1×1 transposed convolution with a stride of 1. Finally, we reduce the feature channels to 1 and apply a sigmoid function to generate a pair of probability masks for each pixel.

Different from existing SISO and MISO DBMs, our decoder network is an MO structure which could further improve the foreground segmentation performance. Furthermore, the advantages of the MO structure are more obvious in scenes with poor light conditions such as nightVideos and lowFramerate in ablation experiment of MO structure (Tab. 5), which is similar to one of the benefits enabled by binocular summation.

B. SOFT FOCAL LOSS

Foreground segmentation suffers from class imbalance in two scenarios. First, foreground objects are too far away from the camera, making them too small in the frame. Second, videos contain a large number of training frames without any foreground object. To prevent vast numbers of easy background examples from misguiding the classifier, multiple modifications are made for the cross entropy (CE) loss function. One of the most remarkable works in this regard is the focus loss [30]. We apply the focal loss proposed for object detection (object-level task) to foreground segmentation (pixel-level task) and redefine it as:

$$p_t = \begin{cases} q_t & y = 1 \\ 1 - q_t & otherwise \end{cases} \tag{1}$$

$$M(p_t) = (1 - p_t)^{\gamma} \tag{2}$$

$$FL = \frac{1}{n} \sum_{t=1}^{n} [M(p_t) \cdot CE_t] \tag{3}$$

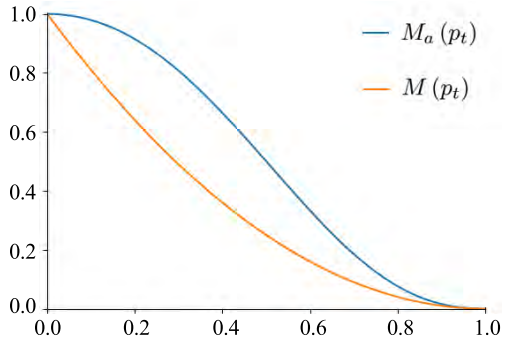


FIGURE 3. A comparison of the *modulating factor* and the *absolute modulating factor*, when $\gamma = 2$.

In the above equation, $y \in \{0, 1\}$ specifies the ground-truth class. $q_t \in [0, 1]$ is the model's estimated probability for example t with label $y = 1$. CE_t represents the cross entropy loss of example t. n is the number of valid pixels in both input frames, excluding pixels in non-region-of-interest (NON-ROI) and unknown region. According to [30], $\gamma \in [0, 5]$ and the ideal value of γ is 2. $M\,(p_t)$ is a *modulating factor* which down-weights the loss assigned to well-classified examples to prevent numerous easy examples from overwhelming the classifier. Furthermore, it is obvious that $(1 - p_t)$ determines whether example t is a well-classified (easy) example or a badly-classified (hard) one. In other words, $(1 - p_t)$ represents the *complexity* of example t. However, we believe the complexity of an example in foreground segmentation, a pixel-level classification task, should be described from two aspects, i.e. complexity (*absolute complexity*) and relative complexity to other examples in the same frame (*relative complexity*). Hence, we extend the *modulating factor* $M\,(p_t)$ to an *absolute modulating factor* $M_a\,(p_t)$ and a *relative modulating factor* $M_r\,(p_t)$.

$$M_a\,(p_t) = (1 - p_t)^{\gamma p_t} \tag{4}$$

$M_a\,(p_t)$ can greatly down-weight the loss of easy examples, playing the role of "rough modulation". We note two properties of it. First, when example t is well-classified and $p_t \to 1$, $M_a\,(p_t)$ goes to 0 and the loss for this example is extremely down-weighted. In this case, $M_a\,(p_t)$ is close to $M\,(p_t)$, especially when p_t is bigger than 0.9 (Fig. 3). Second, as $p_t \to 0$, $M_a\,(p_t)$ is near 1 and the loss is unaffected. Moreover, $M_a\,(p_t)$ can better avoid down-weighting the loss of mis-classified examples (p_t less than 0.5 in Fig. 3) compared with $M\,(p_t)$.

$$M_r\,(p_t) = \frac{e^{1-p_t}}{\sum_{t=1}^{n} e^{1-p_t}} \tag{5}$$

$M_r\,(p_t)$ works as an important complement to $M_a\,(p_t)$, performing "slight modulating" according to *relative complexity*. To avoid "over-modulating", $M_r\,(p_t)$ of different examples should be limited in a small range. Hence, $M_r\,(p_t)$ is defined as normalized exponential *complexity* $(1 - p_t)$. Since e^{1-p_t} is in the interval $(1, e)$, the difference in the *relative complexity* among examples is less than e times. Similar to the Softmax function commonly used in neural networks [44], $M_r\,(p_t)$ shares characteristics such as normalization, interpretability and easy derivation.

To sum up, we term our novel loss function as the soft focal loss (SFL) and define it as:

$$SFL = \sum_{t=1}^{n} [M_a\,(p_t) \cdot M_r\,(p_t) \cdot CE_t] \tag{6}$$

IV. EXPERIMENTS

A. DATASET AND PROTOCOL

We evaluated our method on the CDnet2014 dataset [34], the largest video dataset with pixel accurate groundtruth [9], [15]. The dataset consists of 53 scenes in 11 categories including badWeather (BW), baseline (BL), cameraJitter (CJ), dynamicBackground (DB), interactive Object Motion (IOM), lowFramerate (LF), nightVideos (NV), PTZ, shadow (SH), thermal (TH) and turbulence (TU). It contains 150,000 frames of annotation data, covering a wide range of challenging scenarios. This makes it a rigorous and integrated academic benchmark that allows a more comprehensive assessment of our method.

As officially presented, there are seven evaluation metrics [34], [35]: Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), Precision (Pr) and F-Measure (FM). Among them, *FM* is widely accepted as a metric that can comprehensively represent the overall performance of a model, and it highly relates to the ranking in the CDnet2014 website. Therefore, we primarily use *FM* to compare the performance. Its value ranges from 0 to 1: the larger the value goes, the better the effect is. It is expressed as:

$$FM = \frac{2 \times precision \times recalll}{precision + recalll} \tag{7}$$

where $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$. TP and FP are the *true positive* and *false positive*, FN and TN represent the *false negative* and *true negative*.

B. IMPLEMENTATION DETAILS

We train our X-Net architecture end-to-end using the configurations illustrated in Fig. 2. To incorporate high-level semantic knowledge and improve training efficiency, our encoder network is initialized with the weights of pre-trained VGG-16 model. The experiments are implemented on Keras framework with Tensorflow backend. Data loss is computed using the soft focal loss with $\gamma = 1$ for its best performance. Note that the model does not perform gradient back propagation on the loss caused by NON-ROI regions and the unknown regions during training. RMSProp optimizer is used for updating parameters with an initial learning rate of 1e-4, a batch size of 1 and 60 epochs for each scene's training. To alleviate overfitting, we apply L2 regularization to the

TABLE 2. The test results are obtained by using the same 200 frames and 50 frames from CDnet2014 dataset. Each row shows the average results in each category. Note that only the test frames are included.

Category	Re		Sp		Pr		FPR		FNR		PWC		FM	
	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f	**50f**	**200f**
BW	0.9693	0.9793	0.9999	0.9999	0.9892	0.9884	0.0001	0.0001	0.0307	0.0207	0.0581	0.0292	**0.9790**	**0.9838**
BL	0.9919	0.9969	0.9999	0.9999	0.9971	0.9979	0.0001	0.0001	0.0081	0.0031	0.0261	0.0134	**0.9945**	**0.9974**
CJ	0.9897	0.9939	0.9998	0.9998	0.9950	0.9958	0.0002	0.0002	0.0103	0.0061	0.0631	0.0354	**0.9923**	**0.9948**
DB	0.9915	0.9940	1.0000	1.0000	0.9922	0.9953	0.0000	0.0000	0.0085	0.0060	0.0102	0.0042	**0.9918**	**0.9947**
IOM	0.9858	0.9904	0.9997	0.9997	0.9941	0.9957	0.0003	0.0003	0.0142	0.0096	0.1212	0.0803	**0.9899**	**0.9930**
LF	0.8700	0.9419	0.9998	0.9999	0.8995	0.9203	0.0002	0.0001	0.1300	0.0581	0.0646	0.0244	**0.8840**	**0.9304**
NV	0.9538	0.9830	0.9992	0.9996	0.9636	0.9774	0.0008	0.0004	0.0462	0.0170	0.1613	0.0665	**0.9587**	**0.9801**
PTZ	0.9769	0.9818	0.9999	1.0000	0.9870	0.9846	0.0001	0.0000	0.0231	0.0182	0.0276	0.0098	**0.9819**	**0.9832**
SH	0.9921	0.9968	0.9998	0.9998	0.9945	0.9910	0.0002	0.0002	0.0079	0.0032	0.0482	0.0298	**0.9933**	**0.9938**
TH	0.9820	0.9940	0.9991	0.9992	0.9881	0.9907	0.0009	0.0008	0.0180	0.0060	0.1402	0.0870	**0.9850**	**0.9923**
TU	0.9694	0.9779	0.9998	0.9999	0.9749	0.9816	0.0002	0.0001	0.0306	0.0221	0.0344	0.0211	**0.9721**	**0.9797**
Overall	0.9702	0.9845	0.9997	0.9998	0.9796	0.9835	0.0003	0.0002	0.0298	0.0155	0.0686	0.0364	**0.9748**	**0.9839**

weights of the fusion network and decoder network and set the strength to 2e-4 during training.

Unlike single-image DBM, the X-Net, as a pair-wise input network, needs to select pairs of frames to build the training set. Such networks normally construct a training set after traversing all pairs of given frames. When given m frames, the maximum size of the training set can reach m^2. To take advantage of temporal data without training time explosion, we propose a different strategy and the steps are listed as:

1. Re-arrange the m frames according to their time sequence in the video and re-number them as $1, 2, \cdots, m$.

2. Calculate the serial number difference between paired frames, select those paired frames whose absolute value of difference is less than k_close and greater than 0 to form a training set. For example, when $m = 200$ and $k\_close = 2$, we can select 398 pairs of frames.

3. Randomly split 80% of all selected paired frames for training and 20% for validation.

In the training phase, we set k_close to 6 for 50-training-example case ($m = 50$) and k_close to 2 for 200-training-example case ($m = 200$). Based on this strategy, we choose 470 and 398 pairs of frames respectively, then further split 80% for training and 20% for validation. In 50-frame experiments, each epoch takes about 56 seconds for 320×240 resolution on a single NVIDIA GTX 1080TI GPU during training. In the test phase, our model is fed with consecutive frames pair by pair at the stride of two and can segment around 22 fps for 320×240 resolution on the same GPU.

C. RESULTS

Since the output of our network is two probability masks that value between 0 and 1 for each pixel, we set threshold as 0.5 to convert these probabilities to binary masks for a better explanation. With 200 frames and 50 frames as training examples, we perform experiments on two settings. To make a fair comparison, we use the training examples provided by FgSegNet_S [12], in which examples are selected by random manual selection. Furthermore, we report test results by only considering the frames containing the ground truth labels in Tab. 2. Note that these values are computed using only the test frames, i.e. the training frames are excluded in the performance evaluation.

With the settings mentioned above, the X-Net generates an overall *FM* of 0.9748 with 50-frame experiments and 0.9839 with 200-frame experiments (Tab. 2). As shown in Tab. 2, our method provides high accuracy in foreground segmentation using 200-frames in training. The BL category generates the highest average *FM* compared to the other categories. Though the LF category has the lowest average *FM*, the value also reaches 0.9304. When the number of training examples is downsized to 50 frames, *FM* inevitably decreases by some margins. Especially, in the LF category, *FM* decreases by 0.0464 compared to the model with 200 training examples. However, it still generates acceptable results with an average overall *FM* of 0.9748 across 11 categories, which shows that our method works robustly in many challenging scenarios.

D. COMPARING WITH STATE-OF-THE-ART

We compare our results with six methods mentioned in *Related Works* and the best methods reported on the official website (Tab. 3). FgSegNet_v2, FgSegNet_S and FgSegNet_M (all single-image DBM) are by far the top three DBMs in CDnet2014; in particular, the FgSegNet_M is our baseline method, because the X-Net uses the same multi-scale feature extracting strategy (multi-scale input), backbone network (the first block of VGG-16), and decoder component (transposed convolution); 3D SegNet is a multi-stream fusion DBM with encoder-decoder structure; Cascade CNN is an advanced patch-wise DBM; IUTIS-5 is the best non-deep-learning method. To compare our results with these methods, we need to consider all the frames, i.e. training and test frames, since these methods also include all frames. The *FM* performances of different methods are provided in Tab. 3. In general, DBMs can outperform conventional BMs by large margins, especially in very challenging categories such as nightVideos and PTZ. Furthermore, our model can achieve the highest average accuracy of all. FgSegNet_v2 (currently ranking 1^{st}) has raised the *FM* of FgSegNet_S (ranking 2^{nd}) by 0.0014 with 200-frame experiments (Tab. 4), but

TABLE 3. A comparison of 7 methods considering all the frames in the ground-truths of CDnet2014 dataset. Each row shows the average *FM* of each method. Each column shows the average *FM* of each category. The last row is NON-DBM, others are DBMs. Note that all the other DBMs only take 200 frames in their training sets, while 3D SegNet takes 70% frames for training.

Method	Structure	F-measure											overall
		BW	BL	CJ	DB	IOM	LF	NV	PTZ	SH	TH	TU	
X-Net	MIMO	0.9901	**0.9980**	**0.9975**	**0.9976**	0.9933	**0.9796**	**0.9879**	**0.9945**	0.9956	0.9941	**0.9845**	**0.9920**
FgSegNet_v2 [11]	SISO	0.9900	**0.9980**	0.9961	0.9950	0.9939	0.9579	0.9816	0.9936	0.9966	0.9942	0.9815	0.9890
FgSegNet_S [12]	SISO	**0.9902**	**0.9980**	0.9951	0.9952	**0.9942**	0.9511	0.9837	0.9880	**0.9967**	**0.9945**	0.9796	0.9878
FgSegNet_M [12]	SISO	0.9838	0.9975	0.9945	0.9939	0.9933	0.9558	0.9779	0.9893	0.9954	0.9923	0.9776	0.9865
3D SegNet [15]	MISO	0.9509	0.9691	0.9396	0.9614	0.9698	0.8862	0.8565	0.8987	0.9706	0.983	0.8823	0.9335
Cascade CNN [9]	patch-wise	0.9451	0.9786	0.9758	0.9658	0.8505	0.8804	0.8926	0.9344	0.9593	0.8958	0.9215	0.9272
IUTIS-5 [5]	NON-DBM	0.8289	0.9567	0.8332	0.8902	0.7296	0.7911	0.5132	0.4703	0.9084	0.8303	0.8507	0.7820

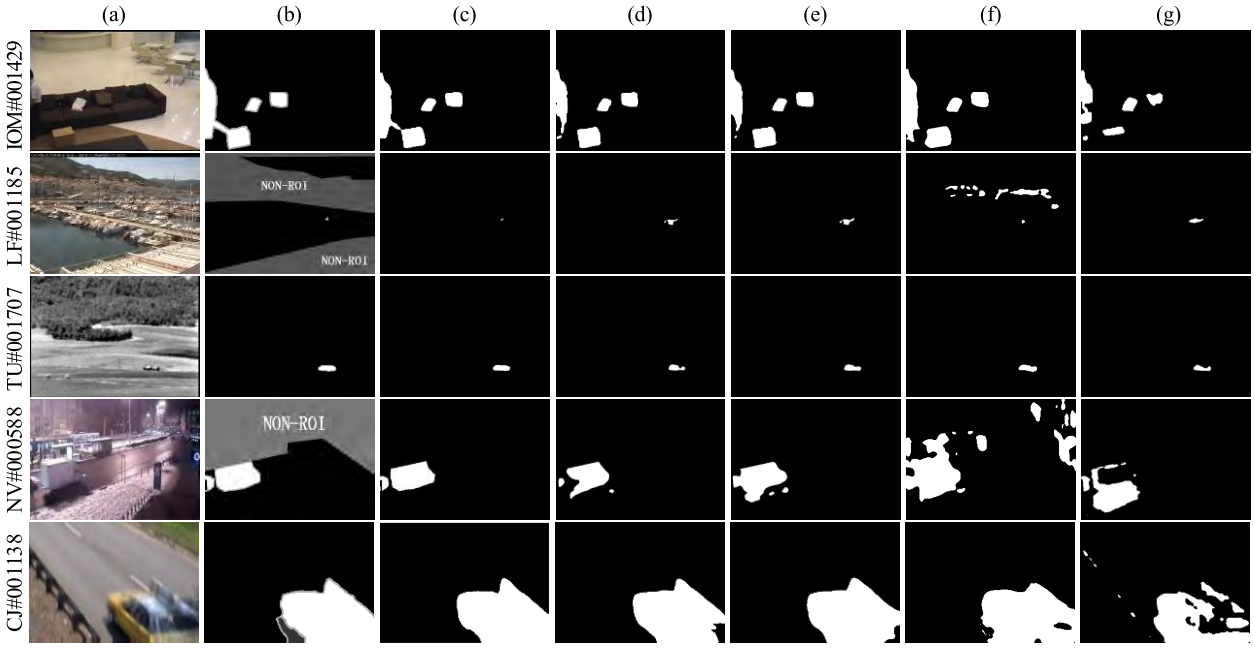


FIGURE 4. Test results on CDnet2014 dataset. (a) input frames, (b) ground-truths, (c) X-Net, (d) FgSegNet_v2, (e) FgSegNet_S, (f) Cascade CNN, (g) IUTIS-5. # (frame number).

TABLE 4. Comparisons between our method and the current top three DBMs in CDnet 2014 benchmark. *FM* is obtained by the same training frames, considering only test frames. The segmentation speed of all methods is based on Keras framework and NVIDIA GTX 1080TI GPU.

	FgSegNet_M	FgSegNet_S	FgSegNet_v2	X-Net
FM(50-frame)	0.9545	0.9633	–	**0.9748**
FM(200-frame)	0.9734	0.9775	0.9789	**0.9839**
speed(fps)	24	34	**38**	22
param.	113M	68M	**67M**	132M

our model can further raise this metric by 0.005, which is 3.5 times higher than the improvement achieved by FgSegNet_v2. In the case of fewer training examples, the advantages of our method are more obvious. As Tab. 4 shows, our method improves overall *FM* by 0.0203 and 0.0115 with 50-frame experiments (Tab. 4) compared with FgSegNet_M and FgSegNet_S respectively. In categories with poor light conditions, such as nightVideos, the *FM* is even raised from 0.9216 [12] to 0.9587 compared with our baseline method FgSegNet_M due to the usage of temporal data.

Due to space limitations, we provide some exemplary results in Fig. 4 that demonstrates the segmentation results of several methods in typical complex scenarios. As can be seen from these figures, our method can accurately estimate the boundaries of objects, both large-scale (Fig. 4 CJ) and small objects (Fig. 4 TU). Meanwhile, our method produces less *false positive* even when facing tiny foreground objects (Fig. 4 LF) and poor illumination (Fig. 4 NV). In addition, for some scenes with great similarities between the foreground and background (Fig. 4 IOM), which causes ambiguity for segmentation, our model can still make an accurate judgment.

However, taking down-scaled frames as input might aggravate the misguiding to the model, which is caused by the NON-ROI regions around the ROI regions, especially in scenes with small ROI regions. On the contrary, the FgSegNet_S and FgSegNet_v2, which are fed with full resolution frames and apply multi-rate dilate convolution to aggregate context. This shows advantages in several categories, such as IOM in which most of its scenes are small ROI region. In addition, multi-scale input strategy is computationally

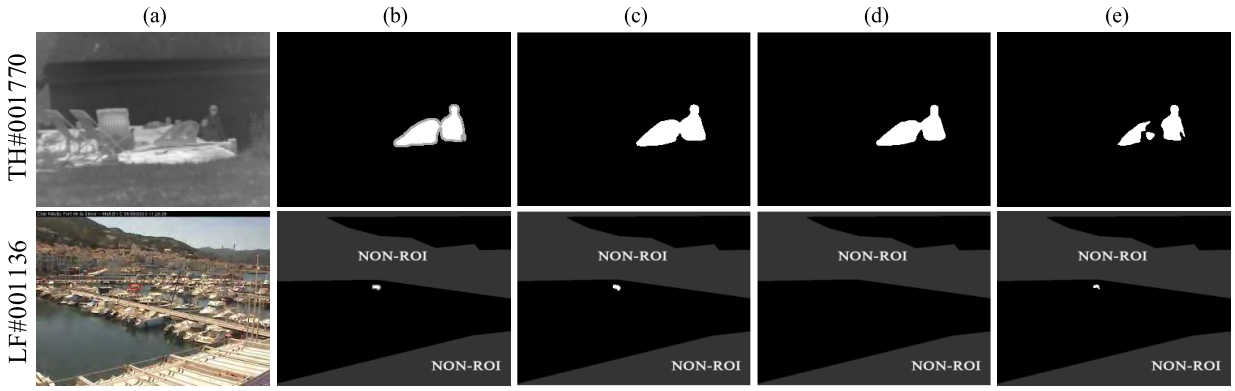


FIGURE 5. A comparison on CDnet2014 dataset. (a) input frames, (b) ground-truths, (c) X-Net with SFL, (d) X-Net with FL, (e) Y-Net with SFL.

TABLE 5. Comparisons between X-Net and its variant, i.e. Y-Net. Both of them are trained with the SFL with 200 frames. Each column shows the average *FM* in each category. Compared with our X-Net, Y-Net only has one branch, but they share the same structures for other parts.

Architecture	BW	BL	CJ	DB	IOM	LF	NV	PTZ	SH	TH	TU	Overall
Y-Net	0.9892	**0.9982**	0.9969	0.9972	0.9916	0.9669	0.9825	0.9937	**0.9974**	0.9875	0.9838	0.9895
X-Net	**0.9901**	0.9980	**0.9975**	**0.9976**	**0.9933**	**0.9796**	**0.9879**	**0.9945**	0.9956	**0.9941**	**0.9845**	**0.9920**

more expensive and tends to cause larger network parameters and slower segmentation speed (Tab. 4). Relevant improvement approaches will be studied in our future work.

V. DISCUSSION

To further evaluate the effectiveness of the MO structure and the soft focal loss in our method, we perform two additional experiments with 200 frames.

A. MO STRUCTURE EXPERIMENTS

Since existing DBMs are either SISO or MISO structures, the ablation experiment of MO structure can only be conducted on the X-Net. We remove one branch of the decoder from the X-Net (Fig. 2) and refer to this multi-input single-output (MISO) network as Y-Net. The comparative experiment is performed by the Y-Net with the soft focal loss. As a result, the X-Net with MIMO structure shows a higher average *FM* than the Y-Net with MISO structure by 0.0025 (Tab. 5). The X-Net has a higher recall in scenes where the environment is camouflaged (TH-(c) & (e) in Fig. 5) or changes dynamically (LF-(c) & (e) in Fig. 5). Especially, the X-Net improves the *FM* over Y-Net by 0.0127 in the LF category. The results reveal that a multi-task learning mechanism brought by the MO structure can straightforwardly facilitate the model to learn the spatio-temporal representation of the background.

B. SOFT FOCAL LOSS EXPERIMENTS

In this study, we perform comparative experiments between the FL and the SFL. Since the ideal value of γ is 2 according to [30], we train the X-Net using SFL and FL with $\gamma = 1, 2, 4$ respectively. As can be seen in Tab. 6, our SFL can further improve the average *FM* by 0.0011. By taking both the *absolute complexity* and *relative complexity* into consideration, the SFL can better focus on hard regions/pixels, such as tiny foreground objects (LF-(c) & (d) in Fig. 5). In contrast, the effectiveness is less obvious in those scenes with moderate-scale foreground objects, such as BL, TH (TH-(c) & (d) in Fig. 5).

TABLE 6. A comparison of *FM* between the SFL and the FL when $\gamma = 1, 2, 4$. The test results are obtained by the X-Net and 200 frames on CDnet 2014 dataset.

	$\gamma = 1$	$\gamma = 2$	$\gamma = 4$	Average
SFL	**0.9920**	**0.9920**	**0.9909**	**0.9916**
FL	0.9913	0.9909	0.9892	0.9905

VI. CONCLUSION

In this work, we propose a novel DBM with MIMO structure for foreground segmentation. To incorporate temporal data, our DBM is designed to be an X-shaped network partially inspired by the human binocular summation mechanism. This model can learn the spatio-temporal representation of the background even using a few training examples. Without any post-processing, our method can achieve state-of-the-art performance on CDnet 2014 dataset. Meanwhile, to overcome class imbalance, we propose a novel soft focal loss by adding the *relative modulating factor* and the *absolute modulating factor* to the cross entropy loss. This strategy can further improve the performance in complex scenes. However, since multi-scale input is computationally more expensive, our future work is to explore an adaptive multi-scale feature extraction network with attention mechanism to boost segmentation speed.

ACKNOWLEDGEMENT

The authors would like to thank Lim et al. for making their FgSegNet code publicly available.

REFERENCES

[1] Z. Luo, P.-M. Jodoin, S.-Z. Li, and S.-Z. Su, "Traffic analysis without motion features," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 3290–3294.

[2] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[3] S. Zhu and L. Xia, "Human action recognition based on fusion features extraction of adaptive background subtraction and optical flow model," *Math. Probl. Eng.*, vol. 2015, Apr. 2015, Art. no. 387464.

[4] Q. Ling, J. Yan, F. Li, and Y. Zhang, "A background modeling and foreground segmentation approach based on the feedback of moving objects in traffic surveillance systems," *Neurocomputing*, vol. 133, no. 10, pp. 32–45, Jun. 2014.

[5] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 914–928, Dec. 2017.

[6] X. Cao, F. Wang, B. Zhang, H. Fu, and C. Li, "Unsupervised pixel-level video foreground object segmentation via shortest path algorithm," *Neurocomputing*, vol. 172, no. 8, pp. 235–243, Jan. 2015.

[7] M. Braham and M. Van Droogenbroeck, "Deep background subtraction with scene-specific convolutional neural networks," in *Proc. IEEE Int. Conf. Syst. Signals Image Process.*, Jul. 2016, pp. 1–4.

[8] 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.

[9] Y. Wang, Z. Luo, and P.-M. Jodoin, "Interactive deep learning method for segmenting moving objects," *Pattern Recog. Lett.*, vol. 96, no. 1, pp. 66–75, Sep. 2017.

[10] W. B. Zheng, K. F. Wang, and F. Y. Wang, "Background subtraction algorithm based on Bayesian generative adversarial networks," *Acta Autom. Sinica*, vol. 44, no. 5, pp. 878–890, Apr. 2018.

[11] L. A. Lim and H. Y. Keles, "Learning multi-scale features for foreground segmentation," 2018, *arXiv:1808.01477*. [Online]. Available: https://arxiv.org/abs/1808.01477

[12] L. A. Lim and H. Y. Keles, "Foreground segmentation using convolutional neural networks for multiscale feature encoding," *Pattern Recognit. Lett.*, vol. 112, no. 1, pp. 256–262, Sep. 2018.

[13] X. Zhao, Y. Chen, M. Tang, and J. Wang, "Joint background reconstruction and foreground segmentation via a two-stage convolutional neural network," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2017, pp. 343–348.

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

[15] D. Sakkos, H. Liu, J. G. Han, and L. Shao, "End-to-end video background subtraction with 3d convolutional neural networks," *Multimedia Tools Appl.*, vol. 77, no. 17, pp. 23023–23041, Sep. 2017.

[16] S. Lian, Z. Luo, Z. Zhong, X. Lin, S. Su, and S. Li, "Attention guided U-Net for accurate iris segmentation," *J. Vis. Commun. Image Represent.*, vol. 56, pp. 296–304, Oct. 2018.

[17] A. C. Sparavigna, "Image segmentation applied to satellite imagery for monitoring water in lakes and reservoirs," *PHILICA*, vol. 1214, pp. 1–5, Jan. 2018.

[18] Y. Hu, A. Soltoggio, R. Lock, and S. Carter, "A fully convolutional two-stream fusion network for interactive image segmentation," *Neural Netw.*, vol. 109, pp. 31–42, Jan. 2019.

[19] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[20] D. Kidd, "The optic chiasm," *Clinican Anatomy*, vol. 27, no. 8, pp. 1149–1158, Nov. 2014.

[21] C. Blakemore, "Binocular depth perception and the optic chiasm," *Vis. Res.*, vol. 10, no. 1, pp. 43–47, Jan. 1970.

[22] R. Blake and R. Fox, "The psychophysical inquiry into binocular summation," *Percept. Psychophys.*, vol. 14, no. 1, pp. 161–185, Feb. 1973.

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4489–4497.

[24] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork convnets for video semantic segmentation," in *Proc. IEEE Eur. Conf. Comput. Vis.*, Nov. 2016, pp. 852–868.

[25] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, "DenseASPP for semantic segmentation in street scenes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3684–3692.

[26] Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep siamese convolutional network for optical aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 10, pp. 1845–1849, Oct. 2017.

[27] H. Spitzer, K. Kiwitz, K. Amunts, S. Harmeling, and T. Dickscheid, "Improving cytoarchitectonic segmentation of human brain areas with self-supervised siamese networks," in *Proc. Int. Conf. Med. Image Comput. Comput. Assist. Interv.*, Sep. 2018, pp. 663–671.

[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: https://arxiv.org/abs/1511.07122

[29] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2097–2106.

[30] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2999–3007.

[31] R. Qin, K. Qiao, L. Y. Wang, L. Zeng, J. Chen, and B. Yan, "Weighted focal loss: An effective loss function to overcome unbalance problem of chest X-Ray14," in *Proc. IOP Conf. Ser., Mater. Sci. Eng.*, Aug. 2018, pp. 012–022.

[32] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," in *Proc. IEEE Int. Conf. Machine Learn.*, Jun. 2013, pp. 231–239.

[33] E. J. Candès, X. D. Li, and Y. Ma, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, May 2011, Art. no. 11.

[34] Y. Wang, P. M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. C. Ishwar, "An expanded change detection benchmark dataset," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 393–400.

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

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent.*, May 2015, pp. 1–14.

[37] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.

[38] Y. Nonaka, A. Shimada, H. Nagahara, and R.-I. Taniguchi, "Evaluation report of integrated background modeling based on spatio-temporal features," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 9–14.

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

[40] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Comput. sci. Rev.*, vol. 11, pp. 31–66, May 2014.

[41] H. Li, Y. F. Zhang, J. B. Wang, Y. L. Xu, Y. Li, and Z. S. Pan, "Inequality-constrained RPCA for shadow removal and foreground detection," *IEICE TRANSACT. Inf. Syst.*, vol. E98-D, no. 6, pp. 1256–1259, Jun. 2015.

[42] M. Braham, S. Pierard, and M. Van Droogenbroeck, "Semantic background subtraction," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 4552–4556.

[43] T. Bouwmans, F. Porikli, B. Hoferlin, and A. Vacavant, "Overview and benchmarking of motion detection methods," *Background Modeling and Foreground Detection for Video Surveillance*, 1st ed. Boca Raton, FL, USA: CRC Press, 2014, ch. 1, sec. 2, p. 2.

[44] Y. I. Bengio, J. Goodfellow, and A. Courville, "Deep feedforward networks," *Deep Learning*, 1st ed. Cambridge, MA. USA, MIT Press, 2016, ch. 6, sec. 2, pp. 180–183.

[45] R. Home, "Binocular summation: A study of contrast sensitivity, visual acuity and recognition," *Vis. Res.*, vol. 18, no. 5, pp. 579–585, Oct. 1978.

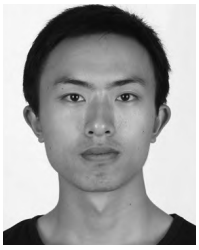
[46] C. Schwarz, S. Manzanera, and P. Artal, "Binocular visual performance with aberration correction as a function of light level," *J. Vis.*, vol. 14, no. 14, p. 6, Dec. 2014.
[47] R. Blakeab and H. Wilson, "Binocular vision," *Vis. Res.*, vol. 51, no. 7, pp. 754–770, Apr. 2011.
[48] A. T. Smith, "Binocular vision: Joining up the eyes," *Current Biol.*, vol. 25, no. 15, pp. R661–R663, Aug. 2015.

JIN ZHANG received the M.S. degree in nuclear science and technology from the Naval University of Engineering, PLA, Wuhan, in 2009. He is currently pursuing the Ph.D. degree in computer science and technology with the Army Engineering University of PLA, Nanjing, China. He is currently a Senior Lecturer with the Army Military Transportation University of PLA, Zhenjiang Campus, Zhenjiang, China. His research interests include computer vision and machine learning.

YANG LI received the B.S. degree from Beihang University, Beijing, China, in 2007, the M.S. degree from the PLA University of Science and Technology, Nanjing, China, in 2010, and the Ph.D. degree from the Army Engineering University of PLA, Nanjing, in 2018, where he is currently an Assistant Professor. His current research interests include computer vision, deep learning, and image processing.

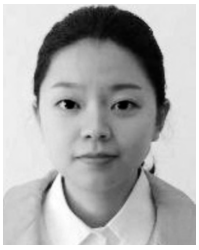
FEIQIONG CHEN received the M.S. degree in computer software and theory from the PLA University of Science and Technology, Nanjing, China, in 2006. She is currently an Assistant Professor with the Army Engineering University of PLA, Nanjing. Her research interests include artificial intelligence and information retrieval.

ZHISONG PAN received the Ph.D. degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2003. He is currently a Professor with the Army Engineering University of PLA, Nanjing. His current research interests include pattern recognition, machine learning, and neural networks.



XINGYU ZHOU received the M.S. degree in information and communication engineering from the PLA University of Science and Technology, Nanjing, China, in 2011. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA. His research interests include computer vision and pattern recognition.

YUDONG LI received the B.S. and M.S. degrees in ship power engineering from the Naval University of Engineering, PLA, Wuhan, in 2008 and 2010, respectively. His research interests include marine auxiliary machinery, and automation and simulation technology.



SHANSHAN JIAO received the B.S. and M.S. degrees in safety engineering from the University of Science and Technology Beijing, Beijing, in 2011 and 2013, respectively. She is currently pursuing the Ph.D. degree in computer science and technology with the Army Engineering University of PLA, Nanjing, China. Her research interests include computer vision and machine learning.