Foreground Gating and Background Refining Network for Surveillance Object Detection

Zhihang Fu, Yaowu Chen, Hongwei Yong, Rongxin Jiang, Lei Zhang, *Fellow, IEEE,*
Xian-Sheng Hua, *Fellow, IEEE*

***Abstract*—Detecting objects in surveillance videos is an important problem due to its wide applications in traffic control and public security. Existing methods tend to face performance degradation because of false positive or misalignment problems. We propose a novel framework, namely Foreground Gating and Background Refining Network (FG-BR Net), for surveillance object detection (SOD). To reduce false positives in background regions, which is a critical problem in SOD, we introduce a new module that first subtracts the background of a video sequence and then generates high quality region proposals. Unlike previous background subtraction methods that may wrongly remove the static foreground objects in a frame, a feedback connection from detection results to background subtraction process is proposed in our model to distill both static and moving objects in surveillance videos. Furthermore, we introduce another module, namely Background Refining stage, to refine the detection results with more accurate localizations. Pairwise non-local operations are adopted to cope with the misalignments between features of original and background frames. Extensive experiments on real-world traffic surveillance benchmarks demonstrate the competitive performance of the proposed FG-BR Net. In particular, FG-BR Net ranks on the top among all the methods on hard and sunny subsets of the UA-DETRAC detection dataset, without any bells and whistles.**

***Index Terms*—Object Detection, Background Subtraction, Pairwise Non-Local Operation, Misalignment, Surveillance Video**

I. Introduction

OBJECT detection is defined by localizing all the objects in an image with tight bounding boxes and simultaneously classifying them into the right categories. It is a fundamental high-level task in many computer vision problems such as object tracking [1], [2] person re-identification [3], [4], object instance segmentation [5], [6] and human action detection [7], [8]. Owing to deep convolutional neural networks (CNNs), we have been witnessing significant advances in object detection in recent years. For generic object detection, CNN-based methods [9]–[13] have achieved remarkable performances on both images and videos.

Detecting objects in surveillance videos, however, still has its unique features challenging the algorithms. False positive on the frame background regions is one of the most critical problems for object detection in surveillance videos. A false positive is a result that indicates a given condition exists, when it does not. In the task of object detection, it refers to that a method incorrectly detects a region as an object with a high confidence score. Because all today's conventional methods [5], [9], [11], [13]–[15] choose Region of Interests (RoI) by sliding window method, false positives on background regions is inevitable. This problem, however, is more serious in surveillance object detection. Backgrounds in surveillance videos change very slowly and a false positive, if it occurs, will exist in the same background area for a while, as shown in Fig. 1(a).

In order to reduce false positives on background regions, many methods [16], [17] subtract the background before detecting objects. The background of a surveillance video can be subtracted by some independent Background Subtraction (BS) methods [18], [19]. But simply implementing BS methods would introduce other issues. The conventional BS methods just subtract the static elements in a video, which means they can falsely eliminate the static foreground objects such as cars and pedestrians waiting in front of the traffic lights. That's why the BS methods are commonly used in the field of moving object detection rather than object detection.

Another problem related to BS process is misalignment. Camera vibration is common in traffic surveillance scenarios due to the complicated situations outdoors. It would lead to misalignments between frames, and further make backgrounds blurred and foregrounds full of noise, as shown in Fig. 1(b). Moreover, since some effective information is removed after BS process (e.g., static foreground objects such as cars and pedestrians), it is not enough to only use the obtained foreground frame as an input to the detection framework. To get a better detection performance, both the original and foreground frames are needed. Thus, misalignment between the feature maps of original and foreground frames also should be considered when using both of them as inputs.

In addition, the object detection in surveillance is a fundamental task. It provides the object information for the subsequent tasks, such as person or vehicle re-identification, vehicle violation lane change detection, which highly depend on the object localization accuracy. Therefore the assessment criteria for evaluating detection methods is much more stringent: the mean Average Precision (mAP) is computed with a higher Intersection over Union (IoU) threshold (0.7) than the conventional one (0.5) in many traffic benchmarks [20], [21]. The high IoU threshold demands high overlaps between detection results and the ground-truth labels, and more efforts are needed to optimize the algorithms for getting high quality bounding boxes.

In this paper, we propose a Foreground Gating and Background Refining Network (FG-BR Net) to accurately detect objects in surveillance videos. The proposed method works on two stages. First, the Foreground Gating (FG) stage supplies high quality RoI proposals by amplifying feature activations on

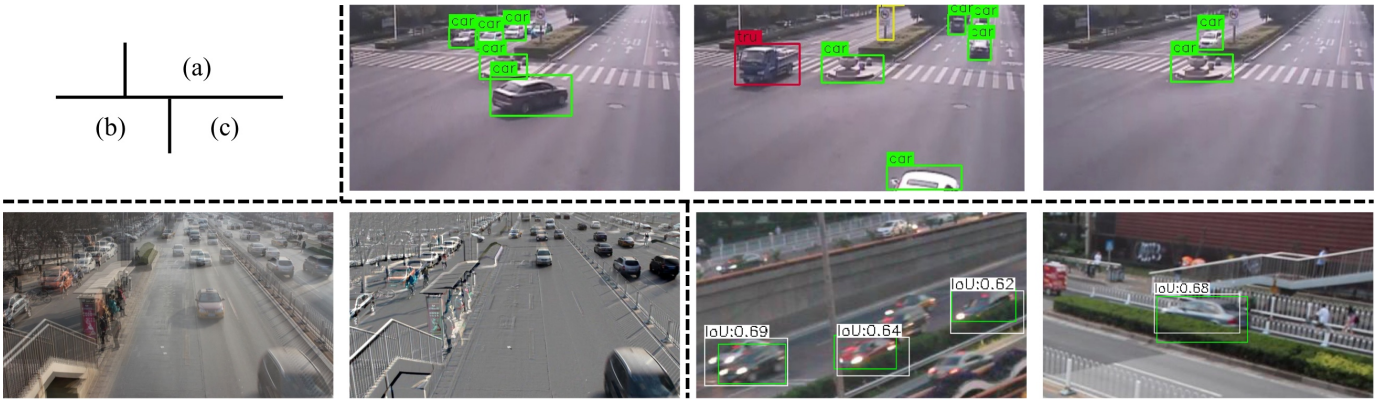


Fig. 1. Challenges on surveillance object detection: (a) false positive problem: the stone in the center of the images has been detected as a car in this video sequence; (b) misalignment problem in Background Subtraction (BS), the left image is a fusion of two frames in different time, showing that the camera has moved slightly, and the right image is the corresponding foreground image using the conventional BS method; (c) the criteria for evaluating detection methods are stringent (IoU $>$ 0.7) in many surveillance datasets.

foreground objects while suppressing background regions (i.e., reducing false positives on these regions). Then, the Background Refining (BR) stage refines those proposals by pairwise non-local operations which pay attention to the background, instead of the frame input itself, to deal with the misalignment problem.

To this end, we set up the Foreground Gating stage with two branches: a multi-layer semantic feature extraction branch which extracts features for proposal generation, and a feature mask branch which first subtracts the backgrounds of current frames and then generates feature-level masks from the foreground frames. Inspired by BS through low-rank subspace exploration [22], we formulate the BS process as a Recurrent Neural Network (RNN) and integrate it into CNN-based network. As a result, the whole stage can be trained end-to-end. We also introduce a feedback connection from detection results to BS process, which helps our BS-RNN block distill both static and moving objects in surveillance videos. And in the Background Refining stage, we recycle the background frames, crop the RoI proposals from both original and background frames and send them as input pairs into a weight sharing convolutional network to get refined detection results. In this stage, pairwise non-local operation is proposed, which is inspired by [23], [24], to cope with the misalignments between features of original and background frames. Briefly, it computes the response at a position of original frames as a weighted average of the features at all positions of background frames.

The contributions of this paper are summarized as follows.

- First, we simplify the state-of-the-art online BS method OMoGMF [22], and formulate it as an RNN to get foregrounds frame by frame. We integrate this BS-RNN into Foreground Gating stage and fine-tune the parameter (ρ) in BS-RNN by high-level detection loss.
- Second, we introduce pairwise non-local operations into Background Refining stage to compute the correlations between the features of original and background frames. As far as we know, we are the first to introduce non-local operations to handle the misalignment problem.
- Finally, we propose a novel FG-BR Net for detecting objects in surveillance videos, which we will demonstrate is accurate and robust. Experiments are conducted on several surveillance datasets, and the FG-BR Net outperforms the state-of-the-art methods. The results are made publicly available.

A preliminary version of this work was published previously [25]. The present work adds to the initial version in significant ways. First, we improve the framework in the previous manuscript by introducing an image transformation to reduce the impact of camera vibration to some extent during the background subtraction process. The modifications of the method made for UA-DETRAC [21] dataset are also provided in this manuscript (Section III). Second, we extend the previous method with a novel Background Refining stage. The new stage further copes with the misalignment problem and enhances the detection performance. Third, considerable new analyses and intuitive explanations are added to the initial results. We also attend the detection competition of UA-DETRAC. Without any bells and whistles, our method ranks on the top among all the methods on hard and sunny subsets. In addition, we compare with a number of recently published methods and confirm that our FG-BR Net still outperforms the existing approaches.

II. Related Work

Surveillance Object Detection. Detecting objects in surveillance videos has its own unique challenges, as we mentioned in Section I. Previous works usually rely on a wide spectrum of analysis tools, from frame differencing [17] to background subtraction [16], to generate semantic features for object detection. These methods mainly focus on moving foreground objects in spite of existence of many static ones that need to be detected such as cars and pedestrians waiting in front of traffic lights at intersections. Object detection in nighttime also imposes additional challenges on those methods for surveillance object detection that should properly deal with over-exposure and defocus aberration. A recent work [26] attempts to solve night object detection problem

by combining HOG and background subtraction. However, it differs from conventional methods by using thermal images as inputs. NoScope [27] proposes an extremely fast framework for surveillance object detection. It is promising to speed up object detection but remains unsatisfactory as it significantly sacrifices its generalization capability to reduce computing costs.

Background Subtraction. The goal of background subtraction is to separate foreground objects from their background in a video sequence. The academic community has achieved fruitful breakthroughs in the field of background subtraction in the past few decades. And several surveys [28]–[30] could be found in literature, providing complete overviews for both novices and experts.

The simplest method only uses a statistic measure, like median [31] or mean [32] over multiple frames to model the static background. Other complex distributions on background pixels, such as MoG [18], are more effective and robust to model slightly changed background. In recent years, online subspace learning approaches have made significant progress on background subtraction from live streams of videos in a real-time online fashion. Several renowned studies [33], [34] focus on this field, among them are GRASTA [19], incPCP [35], OMoGMF [22], ReProCS [36], [37] and MEROP [38]. These online models can greatly speed up the background subtraction through updating the low-rank structure of video background by processing only one frame at a time. They are amenable to efficiently process videos without storing and analyzing a large number of frames.

In OMoGMF, a re-weighted L_2 norm loss function is finally formulated for the tth frame based on current background subspace $\mathbf{U}$ and the coefficient parameter $\mathbf{v}$ for background. Briefly, the loss function minimizes the residual between the tth frame and the corresponding background with a regularization term $\mathcal{R}_B^t(\mathbf{U})$:

$$\begin{aligned} L^t(\mathbf{U},\mathbf{v}) &= ||\mathbf{w}^t \odot (\mathbf{x}^t - \mathbf{U}\mathbf{v})||_2^2 + \mathcal{R}_B^t(\mathbf{U}) \\ \mathcal{R}_B^t(\mathbf{U}) &= \rho \sum_{i=1}^{d} (\mathbf{u}_i - \mathbf{u}_i^{t-1})^T (\mathbf{A}_i^{t-1})^{-1} (\mathbf{u}_i - \mathbf{u}_i^{t-1}) \end{aligned} \tag{1}$$

where $\mathbf{x}^t \in \Re^d$ is a column vector by simply vectorizing the tth frame. d is the number of pixels of the tth frame. $\mathbf{U} \in \Re^{d \times r}$, and each column of $\mathbf{U}$ represents a base vector of background subspace. The background image of the current frame, namely $\mathbf{U}\mathbf{v}$, is formulated as the linear combination of all these base vectors. r is the rank of $\mathbf{U}$ and $\mathbf{v} \in \Re^{r \times 1}$ is the coefficient vector. $\odot$ is element-wise multiplication. $\mathbf{w}^t \in \Re^d$ is the weight vector of the residual between the tth frame $\mathbf{x}^t$ and the corresponding background $\mathbf{U}\mathbf{v}$. It controls the extent to which the residual at each pixel position affects the re-weighted L_2 norm loss. The definition of $\mathbf{w}^t$ depends on the distribution of foreground residual. In OMoGMF, it assumes that the foreground follows a Mixture of Gaussians (MoG) distribution, thus $\mathbf{w}^t$ is related to parameters of MoG and residual. In this paper we will make a simple formulation for $\mathbf{w}^t$. $\mathcal{R}_B^t(\mathbf{U})$ is the background subspace regularization term, ρ is a regularization parameter which controls the strength of impact that the previous frames make on current one. $\mathbf{u}_i$ and $\mathbf{u}_i^{t-1}$ denote the row vector of $\mathbf{U}$ and old background subspace $\mathbf{U}^{t-1}$ respectively, and $\{\mathbf{A}_i^t\}_{i=1}^d$ is an auxiliary variable. Equation (1) can be solved by the following re-weighted iterative algorithm [39] through solving each iteration.

$$\mathbf{v}^t = ({\mathbf{U}^{t-1}}^T diag(\mathbf{w}^t)^2 \mathbf{U}^{t-1})^{-1} {\mathbf{U}^{t-1}}^T diag(\mathbf{w}^t)^2 \mathbf{x}^t \tag{2}$$

The closed-form solution for $\mathbf{U}^t$ for $i = 1, \cdots, d$ is:

$$\begin{aligned} \mathbf{A}_i^t &= \frac{1}{\rho}\left(\mathbf{A}_i^{t-1} - \frac{{w_i^t}^2 \mathbf{A}_i^{t-1}\mathbf{v}^t{\mathbf{v}^t}^T\mathbf{A}_i^{t-1}}{\rho + {w_i^t}^2 {\mathbf{v}^t}^T \mathbf{A}_i^{t-1}\mathbf{v}^t}\right); \\ \mathbf{b}_i^t &= \rho \mathbf{b}_i^{t-1} + {w_i^t}^2 x_i^t \mathbf{v}^t. \\ \mathbf{u}_i^t &= \mathbf{A}_i^t \mathbf{b}_i^t \end{aligned} \tag{3}$$

where $\{\mathbf{b}_i^t\}_{i=1}^d$ is another auxiliary variable for recurrently updating $\mathbf{U}^t$ together with $\{\mathbf{A}_i^t\}_{i=1}^d$. Finally $\mathbf{U}^t\mathbf{v}^t$ is the background of the tth frame x^t.

In addition, transformed-OMoGMF [22] has introduced an image transformation operator to mitigate the effects of video shakes. The parameters of the transformation can be obtained by optimizing the objective function. Due to the complicated installation situation in the outdoor, camera vibration is common in traffic surveillance scenarios. Image transformation needs to be considered in response to the camera vibrations.

However, there exists flaws when simply shifting this method to detection problem for pixel-level mask generation. For example, it cannot handle the situation where target objects stop moving for a while in videos. We address this problem with a close-loop pipeline by feeding back object detection results to model static foreground objects.

Non-Local Operation. The method non-local means [40] was originally proposed for image denoising. It is based on a non-local averaging of all pixels in an image and allows distant pixels to contribute to the filtered response at a location based on patch appearance similarity. Subsequently, several elegant methods share the non-local matching insights in other research fields such as super-resolution [41] and image restoration [42]. The self-attention method for machine translation in [24] computes the response at a position as a weighted average of correlations at all positions in a sequence. As discussed in [23], the self-attention can be viewed as a form of the non-local means operation. Besides, the work [23] also proposes a non-local block based on neural networks. It computes the response at each position of CNN feature layers rather than the image pixels. It has achieved great improvements on the task of video classifications.

III. Approach

In this section, we present the FG-BR Net to enable effective object detection for surveillance videos. The framework consists of two stages: 1) a Foreground Gating stage which supplies high quality RoI proposals by amplifying feature activations on foreground objects while suppressing background regions; 2) a Background Refining stage which handles the misalignment between features of backgrounds and original frames, and refines those proposals by pairwise non-local weighted background fusions.

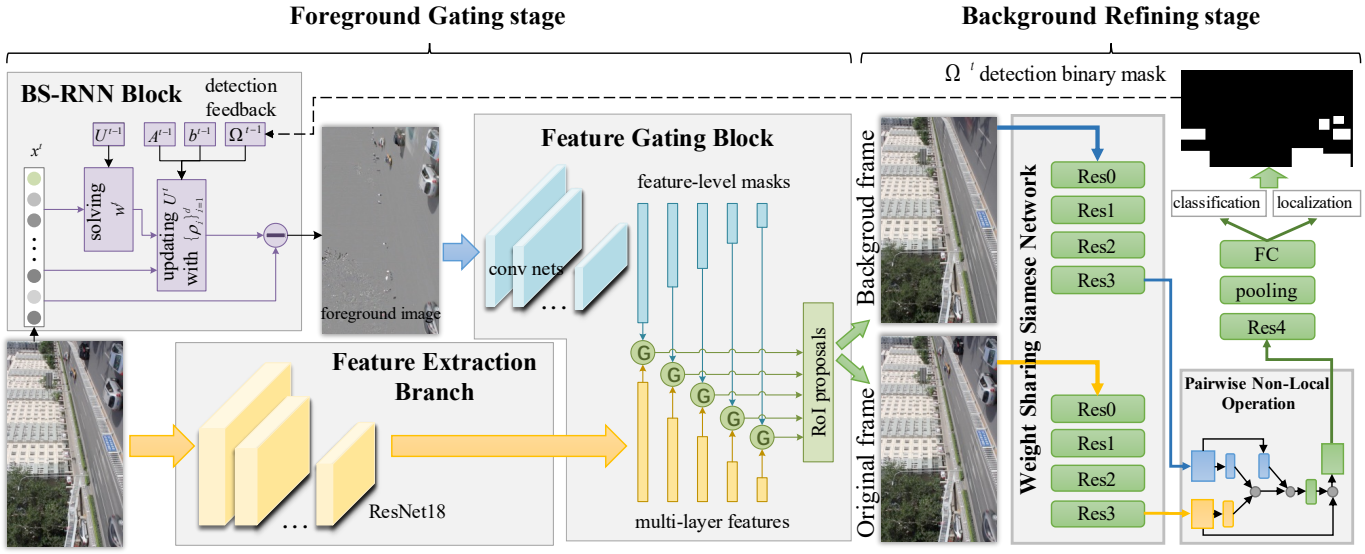


Fig. 2. Pipeline of our FG-BR Net. In FG stage, a frame is fed to two branches in parallel. In BS-RNN block, the frame is vectorized as a column vector x^t first. $w^t, \{\mathbf{A}_i^t\}_{i=1}^d, \{\mathbf{b}_i^t\}_{i=1}^d$ and $\mathbf{U}^t$ are updated in turn based on $\mathbf{U}^{t-1}, \{\mathbf{A}_i^{t-1}\}_{i=1}^d, \{\mathbf{b}_i^{t-1}\}_{i=1}^d$ and the feedback masks Ω^{t-1} from the $(t-1)$th frame. In BR stage, the backgrounds from BS-RNN block is combined with the original patch as the input pair. The pairwise non-local operation is embedded before the last residual block (Res4). Time delay is indicated with dash line.

A. Foreground Gating stage

1) Feature-level Mask Generation Branch: Masking mechanism has been proved effective on many computer vision tasks such as semantic segmentation [43] and image classification [44]. For object detection, we experimentally show in this paper that the masking operation also helps object detection and it is sensitive to the inputs. We introduce a BS-RNN block and a feature gating block which distills both static and moving foregrounds and reduces false positives on background regions respectively.

BS-RNN Block. OMoGMF [22] constructs a background subspace $\mathbf{U}^t$ with the rank of r, and a learnable coefficient parameter $\mathbf{v}^t$ to handle the cases with fast changes in luminance. We simplify this method and formulate it as an RNN to explore the temporal information.

OMoGMF models the distribution of input x^t to solve the weight parameter $\mathbf{w}^t$ in (1). The method in OMoGMF is intricate and does not fit for our detection network. Based on the principle proposed in OMoGMF that w_i^t should be large to increase its impact on loss function where the residual, namely $|x_i^t - \mathbf{u}_i^{t-1^T}\mathbf{v}^{t-1}|$, is small, we reformulate $\mathbf{w}^t$ as following:

$$w_i^t = \frac{1}{|x_i^t - \mathbf{u}_i^{t-1^T}\mathbf{v}^{t-1}|^p + \varepsilon} \tag{4}$$

where ε is set to be a very small number. The performance is not sensitive to p ranging from 0.5 to 1 in our experiments and we set L_p norm as $p = 1$ for convenience. Equation (4) keeps the core property that w_i^t should be large where the residual is small.

In real applications, the subspace $\mathbf{U}^t$ only changes slightly over time, thus we set r to a small value, even to rank-one, which still works well in our experiments. So the vector $\mathbf{v}^t \in \Re^r$ degenerates into a real number and we set it to 1 for simplification. Besides, fixing $\mathbf{v}^t$ to 1 in this paper is also based on the precondition that luminance in surveillance scenarios usually changes slowly and it can be handled by updating $\mathbf{U}^t$ frame by frame. Due to $r = 1$, $\mathbf{A}_i^t \in \Re^{r\times r}$, $\mathbf{b}_i^t \in \Re^r$ and $\mathbf{u}_i^t \in \Re^r$ all degenerate into real numbers A_i^t, b_i^t, u_i^t for $i = 1, \cdots, d$.

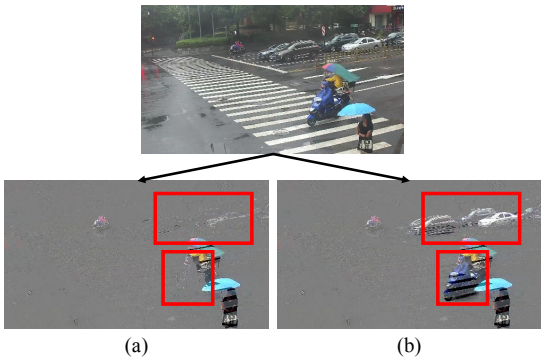


Fig. 3. Foreground frame by (a) OMoGMF [22]; (b) BS-RNN with detection result feedbacks. Static cars and cyclists are preserved as foreground by feedbacks.

We further parameterize the regularization parameter $\rho \in \Re$ to multiple $\{\rho_i\}_{i=1}^d \in \Re^d$ which could fine-tune outputs of BS-RNN through different RGB channels and positions. The corners in surveillance videos, for example, almost remain unchanged compared with the center and thus a larger ρ_i is needed in these corner areas. We automatically fine-tune $\{\rho_i\}_{i=1}^d$ in the FG stage.

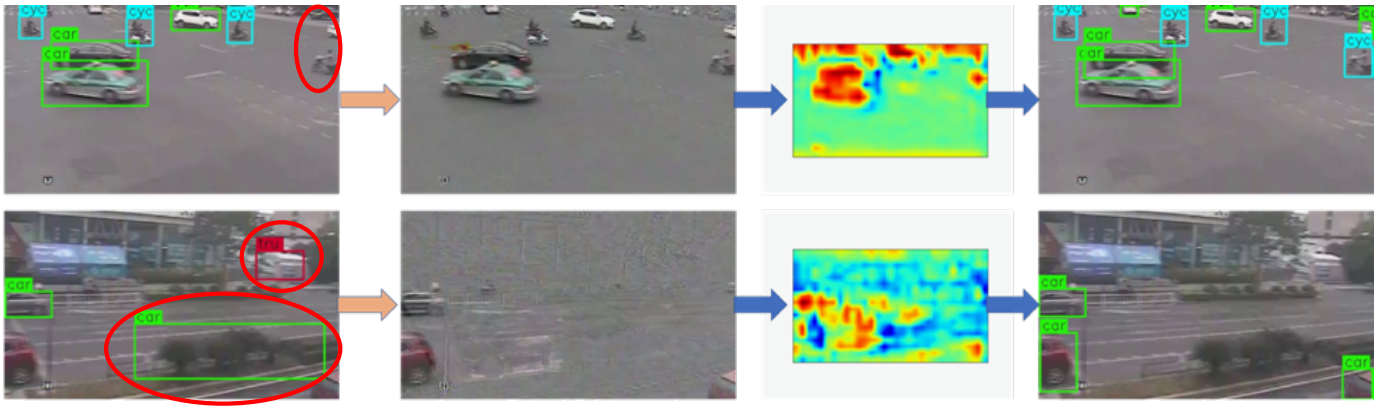


Fig. 4. Explanation of how Foreground Gating stage helps detection. Enhanced by feature masks of the foreground frame, the red circle regions in the left image are finally detected or eliminated. For a simple visualization, we select only one channel in 1/16 (conv4) feature layer and up-sampling it by bilinear interpolation. A bounding box is plotted if its confidence score is larger than 0.3.

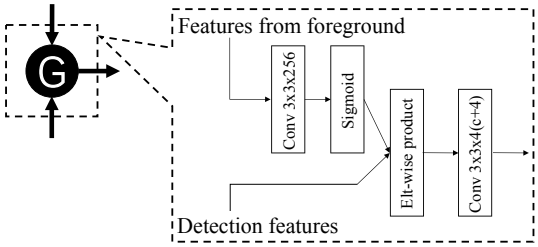


Fig. 5. Details of a gate module. The number of output channel is $4\times(c+4)$. c denotes class number, the first 4 is anchor number and the second denotes 4 coordinates for localization.

In short, Equation (2) and (3) can be simplified as following:

$$\begin{aligned} w_i^t &= \frac{1}{|x_i^t - u_i^{t-1}| + \varepsilon} \\ A_i^t &= \frac{A_i^{t-1}}{\rho_i + {w_i^t}^2 A_i^{t-1}} \\ b_i^t &= \rho_i b_i^{t-1} + {w_i^t}^2 x_i^t \\ u_i^t &= A_i^t b_i^t \end{aligned} \tag{5}$$

We just need compute $\mathbf{U}^t$ through updating $\mathbf{w}^t$, $\mathbf{A}^t$ and $\mathbf{b}^t$ in a recurrent fashion and no longer have to iteratively solve $\mathbf{v}^t$. $\mathbf{U}^t$ turns out to the background of current frame x^t as $\mathbf{v}^t$ is set to be 1. The above iterative computing can be implemented recurrently in an RNN structure, and thus we call it BS-RNN block. This network view of the BS-RNN block allows us to train it together with the feature extraction and detection branches in an end-to-end fashion.

In addition, as illustrated in Fig. 2, one of the most distinctive features in the proposed stage is that we allow the high-level detection results to feed back into the BS-RNN block to further improve its accuracy. Denote by $\Omega^{t-1} \in \Re^d$ the binary masks that are complement to the masks of detected objects for the previous frame x^{t-1}:

$$\Omega_i^{t-1} = \begin{cases} 0, & i \in \Theta^{t-1} \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where Θ^{t-1} denotes the set of indices of pixels which locate in the detection bounding boxes in the $(t-1)$th frames. Then, instead of updating $\{A_i^t\}$ and $\{b_i^t\}$ for all $i = 1, \cdots, d$, we only update those background pixels for $\Omega^{t-1} = 1$. Feeding back the object detection results into BS-RNN plays an important role in our application, as it could avoid neglecting static foreground objects of interest (e.g., the cars waiting for traffic lights).

With these feedbacks, the BS-RNN block is able to distill both static and moving objects in surveillance videos. Fig. 3 shows how the feedbacks of object detections successfully correct the errors in BS-RNN block.

Feature Gating Block. With the above BS-RNN block, foreground masks are extracted from video frames at pixel level. These foregrounds are then fed into a feature gating block to generate feature masks.

As shown in Fig. 2, the feature gating block consists of hierarchical feature layers with connected gate modules. Since we use multi-layer features to detect objects at different scales, namely multi-scale detector, the layers in feature gating block are extracted to generate corresponding feature-level masks. The role of feature-level masks is to assist the feature extraction branch in further optimizing its feature activations: amplifying foreground regions with large mask values, while suppressing the activations in background regions with small mask values. The feature-level masks are generated from the foreground frame, in which the disturbance that causes false positives in background regions has been already subtracted. Intuitive examples are shown in Fig. 4. Since most of image regions are suppressed as background, the feature map can be quickly down-sampled to $1/8$ of input size by convolutional layers with a stride of 2. Then we continue to perform down-sampling until the feature size reduces to $1/128$ of input size. To balance between computation cost and detection accuracy, the channel number is set to 32 all through the down-sampled layers.

As shown in Fig. 5, the gate modules in the feature gating block generate feature-level masks by using sigmoid activation functions, which are used to suppress background regions of feature maps for detection. Although binary masks work well

on semantic segmentation [43], we experimentally show that feature-level masks are much more proper for object detection. An ablation study is performed in Section IV-B.

2) Semantic Feature Extraction Branch: We use ResNet18 [45] as the pre-trained model to extract multiple feature layers for object detection. Similar to SSD, we remove the average pooling layers from the pre-trained model and add auxiliary convolutional layers to detect large sizes of objects. The sizes of feature map for object detection are $1/8$ (conv3), $1/16$ (conv4), $1/32$ (conv5), $1/64$ (conv6) and $1/128$ (conv7) of the input frames, both in width and height. At each layer, we consider only 4 anchor bounding boxes with different scales and ratios at each location. The number of anchors is reduced to 4 to save memory and cut down complexity. We adopt cross-entropy loss and smooth L_1 loss to jointly train object classification and bounding box regression.

Although many recent researches [46], [47] show that top-down structures do boost detection performance especially for small objects, we do not adopt this structure for two reasons: 1) pursuit of efficiency and 2) fair comparison with the baseline methods. It will not be surprising for getting a better result after adding a top-down structure in FG stage, but that is beyond the scope of this paper.

B. Background Refining stage

Single stage methods [9], [10] usually under-perform two stage methods [5], [15], [46], especially in large scale datasets, such as COCO [48]. Because the features obtained by RoI pooling operation are more accurate in the two stage methods, the predictions for classification and localization are both refined in the second stage. The single stage methods do not have this refinement process.

Moreover, the misalignment is a crucial problem when using the foreground or background as a gate for the original frame. There are two main reasons for this problem. One is camera vibration, which leads to the misalignment between frames in pixel level, and further makes the background subspace blurred, as illustrated in Fig. 1(b). The other one is unshared parameters between two branches of FG stage, which leads to the slightly misalignment in feature level.

Based on the above two defects, we introduce a Background Refining stage as the second stage to complete the detection framework. Similar to R-CNN [12], we first crop the RoI proposals, which are generated in the FG stage, from both original images and backgrounds. These cropped original and background patches are grouped into pairs, then sent into a Siamese-like network, which are two ResNet50, sharing the same weight parameters during training and inference. In this stage, we handle the misalignment problem by pairwise non-local operations between the original frames and its backgrounds. Instead of the pixel-wise multiplication, which is sensitive to misalignments, we compute the response at a position in feature maps of the original frames as a weighted average of the features at all positions in feature maps of the background frames. Every position in background features is considered when computing the response at a position of original frames.

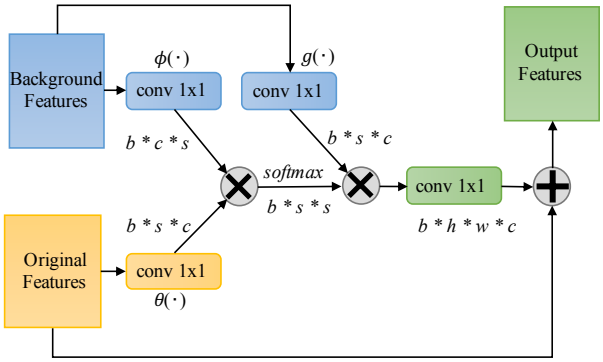


Fig. 6. Details of pairwise non-local operations. $b*h*w*c$, namely $batchsize*height*width*channel$, indicates the shape of data tensor. For the convenience of matrix operations, we reshape the input tensors to $b*s*c$, where $s=h*w$, and permute the background tensor to $b*c*s$. $\otimes$ denotes matrix product of two tensors and $\oplus$ denotes element-wise sum.

Pairwise Non-Local Operation. Following the non-local mean operation in [40], Wang *et al.* [23] defined a generic non-local operation in deep neural networks as:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\mathbf{x})}\sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) g(\mathbf{x}_j) \tag{7}$$

where $\mathbf{x}$ is the input vector and $\mathbf{y}$ is the output. i is the index of the target position whose response is to be computed. $\mathcal{C}(\mathbf{x})$ is a factor that normalize the output. The function $f(\cdot)$ computes the weight, which represents the relationship between i and all j (all elements in $\mathbf{x}$). Then with the representations of the input vector, namely $g(\mathbf{x}_j)$, the output response is obtained as a weighted sum of all positions of the input vector.

In machine translation field, Vaswani *et al.* [24] also introduced a self-attention module to draw global dependence between input and output. The self-attention function is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{8}$$

where matrices Q, K and V denote a set of queries, keys and values respectively, and d_k denotes the dimension of keys. The self-attention module is described as mapping a query and a set of key-value pairs to an output. The output is also computed as a weighted sum of the values. As pointed in [23], self-attention module can be seen as a specific form of non-local means operation when in (7):

$$\begin{aligned} f(\mathbf{x}_i, \mathbf{x}_j) &= e^{\theta(\mathbf{x}_i)^T \phi(\mathbf{x}_j)} \\ \mathcal{C}(\mathbf{x}) &= \sum_{\forall j} f(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{9}$$

The self-attention [24] is reflected in that fact that the output of a word is related to all words in the sequence it belongs to (the sequence itself). Similarly, the output response in non-local operation [23] is also computed as a weighted average of all elements in the signal (image, sequence, video) itself.

Contrast to the mechanism of *attention to itself*, we modify (7) and introduce a pairwise non-local operation to let the output response pays *attention to its background*:

$$\mathbf{y}_i = \frac{1}{\mathcal{C}(\hat{\mathbf{x}})} \sum_{\forall j} f(\mathbf{x}_i, \hat{\mathbf{x}}_j) g(\hat{\mathbf{x}}_j) \tag{10}$$

where $\mathbf{x}$ denotes the features of an image and $\hat{\mathbf{x}}$ denotes the features of its background. Instead of the pixel-wise multiplication in the FG stage, all positions $(\forall j)$ in the background are considered when computing the response at any position in this BR stage. The pairwise non-local operation would alleviate the impact of misalignment by taking all positions of background into consideration and reassigning the weights based on similarity. Fig. 6 shows the pairwise non-local operations in detail.

For simplicity, we follow the settings in [23]: $\mathcal{C}(\cdot)$ and $f(\cdot)$ are defined as in (9), and $g(\cdot)$ is defined as a linear embedding:

$$g(\hat{\mathbf{x}}_j) = W\hat{\mathbf{x}}_j \tag{11}$$

where W is a weight parameter set of 1×1 convolution. Both $\theta(\cdot)$ and $\phi(\cdot)$ in (9) are linear embeddings, similar to (11).

The residual connection is implemented following the non-local network [23] and self-attention [24]. When the pair of positive samples (the patch which contains a car for example) are sent to the pairwise non-local operation block, the output responses tend to be small in the main areas because the background patches have no similarity. But when the pair belongs to negative sample, the responses tend to be large as the most areas of the original image patch are backgrounds. With the residual connection, the large responses would counteract those from original image patch and eventually suppress the classification confidence score of this patch.

C. Algorithm Base for BS-RNN

It is worth noting that there are three main reasons why we choose OMoGMF [22] as the base for BS-RNN block: 1) OMoGMF is effective on background subtraction and has a better performance than other online models such as GRASTA [19], incPCP [35] and ReProCS [36], [37], as pointed out by Yong *et al.* [22]. 2) OMoGMF is friendly to GPU migration because it updates the background frame through a pixel-wise way without singular value decomposition (SVD). 3) It is convenient to formulate OMoGMF as an RNN for deep coupling with CNN-based object detection network. In contrast, GRASTA, incPCP and ReProCS all need SVD, which is difficult to perform gradient back propagation.

Meanwhile, many deep learning based background subtraction works [49]–[51] have been introduced in recent years. Bouwmans *et al.* [51] reported that the top current background subtraction methods are based on deep neural networks with a large gap of performance in comparison on the conventional unsupervised approaches. These deep learning approaches are supervised methods and rely on explicit foreground annotations for training. However, we focus on object detection in this paper and annotations of the datasets used in our experiments are class-specific bounding boxes instead of foreground pixels. As a result, the unsupervised OMoGMF is a reasonable choice of the base for BS-RNN block.

D. Implementation Details

Image Transformation Operator $\boldsymbol{\tau}$. As demonstrated in Section III-B, camera vibrations would lead to misalignment problem in pixel level. We introduce an image transformation operator in BS-RNN block to reduce the influence of camera vibrations and further obtain clearer foreground and background frames. Owing to t-OMoGMF [22], an image transformation operator can be introduced to the model to make all the frames aligned to a central position. The loss function should be changed as:

$$L^t(\mathbf{U}, \mathbf{v}, \tau) = ||\tau \cdot \mathbf{x}^t - \mathbf{U}\mathbf{v}||_p^p + \mathcal{R}_B^t(\mathbf{U}) \tag{12}$$

where τ is the parameter of the image transformation operator and $\tau \cdot \mathbf{x}$ denotes the transformation with parameter τ works on the image $\mathbf{x}$. The τ can be obtained by optimizing the follow

$$L^t(\mathbf{U}, \mathbf{v}, \Delta\tau) = ||\tau \cdot \mathbf{x}^t + \mathbf{J}\Delta\tau - \mathbf{U}\mathbf{v}||_p^p + \mathcal{R}_B^t(\mathbf{U}) \tag{13}$$

where $\Delta\tau$ is a shift variable of τ, and $\mathbf{J}$ is Jacobian matrix of τ w.r.t. $\mathbf{x}^t$ and τ can be updated by the equation $\tau = \tau + \Delta\tau$. In addition, image pyramid strategy is adopted to accelerate computation.

Fine-tune the Parameter $\boldsymbol{\rho}$. For the FG stage, an intractable problem during training is that all frames have to be sent as inputs chronologically when our BS-RNN block is embedded into network, because the auxiliary variables $\mathbf{A}^t$ and $\mathbf{b}^t$ have to be updated frame by frame successively. When the video sequences are too long, training them in chronological order would hurt the capacity of $stochastic$ gradient descent (6-8 point decline in our experiments).

To address this issue, we propose a new way to train our FG stage. First we set all $\{\rho_i\}_{i=1}^d$ to 0.9 as initial values, get foregrounds of all frames chronologically and save them into a cache, and then we randomly choose frames with their pre-saved foregrounds to train the whole nets excluding $\{\rho_i\}_{i=1}^d$ with SGD, batch size 4 for 40k iterations. Second we begin to fine-tune $\{\rho_i\}_{i=1}^d$ through the backward gradients from the detection loss. During this period the inputs are chronologically ordered. After 3k iterations we update the cache with new parameter $\{\rho_i\}_{i=1}^d$. Then we repeat the first step for 3k iterations and next repeat in this way. We fine-tune $\{\rho_i\}_{i=1}^d$ from a coarse initial value 0.9. It could also be regard as that the unsupervised problem (background subtraction) is supervised by the high-level detection loss.

Training Details. We train the FG stage using SGD with an initial learning rate 0.001, 0.9 momentum, 0.0005 weight decay, in a batch size of 4. In training phase, ground truth bounding boxes are used to form masks fed back into the BS-RNN. Detection results with confidence scores larger than 0.3 are used to generate masks in the inference. The Non-Maximum Suppression (NMS) threshold is set to be 0.65 to ensure the proposals have a large recall rate. The FG stage can work as a detector, by switching the class-agnostic classification to class-specific and setting the NMS threshold to be 0.5 in our experiments. We treat the FG stage as a detector for ablations. The object detection accuracy is measured by Average Precision (AP) and mean Average Precision (mAP).

The Background Refining stage is trained similar to R-CNN [12]: crop and resize the proposals to 224×224, and

TABLE I
SINGLE-MODEL DETECTION RESULTS ON TSD TEST SET. OBJECT CATEGORIES INCLUDE CAR, BUS, TRUCK, CYCLIST AND PEDESTRIAN. DARKNET19 IS USED AS BACKBONE IN YOLOV2, AND VGG16 IS USED IN SSD, MS-CNN AND FG-VGG16. THE DEFAULT FG IN TABLE IS BUILT ON RESNET18. THREE VERSIONS OF MS-CNNS DIFFER FROM EACH OTHER IN THE SIZE OF INPUT IMAGES AND NETWORK DESIGNS [13]. THE AVERAGE PRECISION (AP) ON EACH CATEGORY IS OBTAINED WITH AN IOU THRESHOLD 0.5.

method	test set for the-same-scenario (test set 1)						test set for cross-scenario (test set 2)					
	car	bus	tru	cyc	ped	mAP	car	bus	tru	cyc	ped	mAP
YOLOv2 [10]	61.0	71.2	59.1	66.8	44.2	60.5	58.8	49.4	45.7	49.4	23.2	45.3
SSD [9]	83.1	77.0	66.3	74.9	51.1	70.5	74.1	**61.7**	**59.6**	63.1	33.2	58.3
MS-CNN v1 [13]	86.4	60.5	58.8	67.7	46.5	64.0	68.3	36.1	39.4	54.9	36.3	47.0
MS-CNN v2 [13]	87.0	70.2	66.5	73.4	52.4	69.9	70.7	42.0	52.3	62.2	41.9	53.8
MS-CNN v3 [13]	87.3	73.3	67.5	74.7	57.7	72.0	70.2	40.7	44.8	61.7	40.7	51.6
FG-ResNet18	89.8	77.0	69.4	75.0	58.9	74.0	70.0	61.6	55.7	62.8	42.8	58.6
FG-VGG16	**91.2**	**81.2**	**72.3**	**79.8**	**63.3**	**77.5**	**74.4**	54.7	54.9	**69.9**	**44.7**	**59.8**

TABLE II
ABLATION STUDY ON TSD TEST SET. FOREGROUND AND SUBNET DENOTE THE USE OF FOREGROUNDS AND MASK GENERATION SUBNET RESPECTIVELY, AND ELE-WISE IN ρ MEANS USE OF $\{\rho_i\}_{i=1}^{d}$ FINE-TUNED BY DETECTION LOSS.

model	foreground?	subnet?	mask	ρ	test1	test2	S	M	L	adverse scenarios
A			pixel-level	ele-wise	66.5	50.7	20.0	64.7	75.1	46.6
B			N/A	N/A	71.6	52.5	30.2	67.3	73.5	48.9
C	✓		N/A	0.9	70.5	53.5	32.1	67.7	71.4	54.2
D		✓	feature-level	N/A	73.5	55.1	31.7	69.1	74.8	51.6
E	✓	✓	feature-level	0.9	72.8	58.0	32.5	70.1	76.3	55.8
F	✓	✓	feature-level	ele-wise	**74.0**	**58.6**	**34.7**	**70.7**	**76.5**	**56.2**

TABLE III
RESULTS ON TSD TEST SET. THE FIRST THREE COLUMNS ARE AP ON DIFFERENT CATEGORIES OF OBJECT SIZES. S DENOTES SMALL SIZE OBJECTS (HEIGHTS <20 PIXELS), M DENOTES MEDIUM SIZE OBJECTS (HEIGHTS BETWEEN 20 AND 52 PIXELS) AND L DENOTES LARGE SIZE OBJECTS (HEIGHTS >52 PIXELS). THE LAST COLUMN IS MAP IN ADVERSE SCENARIOS FOR DETECTION DUE TO OVER-EXPOSURE AT NIGHT AND MOTION BLURS CAUSED BY CAMERA VIBRATIONS.

method	input size	AP on size			mAP on adverse scenarios
		S	M	L	
YOLOv2	416 × 416	12.6	55.6	71.3	42.5
SSD	300 × 300	15.9	**72.8**	73.4	52.6
MS-CNN v1	682 × 384	**55.7**	66.7	44.9	38.5
MS-CNN v2	1024 × 576	51.1	71.0	56.5	46.9
MS-CNN v3	1356 × 768	50.5	70.3	61.4	46.8
FG	496 × 279	34.7	70.7	**76.5**	**56.2**

TABLE IV
ABLATION ON UA-DETRAC VALIDATION SET. "NON-LOCAL" COLUMN INDICATES THE TYPE OF NON-LOCAL OPERATIONS THE MODEL INVOLVES. "POS:NEG" COLUMN INDICATES THE RATIO BETWEEN POSITIVE AND NEGATIVE PROPOSALS IN A MINI-BATCH. "AP@0.5" MEANS AVERAGE PRECISION THAT OBTAINED WITH THE IOU THRESHOLD 0.5 AND "AP@0.7" MEANS AP WITH IOU THRESHOLD 0.7.

model	non-local	backbone	pos:neg	AP@0.5	AP@0.7
baseline				82.66	76.67
A		ResNet18	1:1	85.05	79.12
B	normal	ResNet18	1:1	85.93	80.30
C	pairwise	ResNet18	1:1	86.31	81.85
D	pairwise	ResNet50	1:1	**87.65**	**83.02**
E	pairwise	ResNet50	1:4	87.32	82.76

train the Siamese-like network from ResNet50 pre-trained on ImageNet. The cross-entropy loss and smooth L_1 loss are still adopted to jointly train object classification and bounding box regression in BR stage. The pairwise non-local block is added to right before the last residual block of a stage, to balance the spatial information and semantic information. The BR stage is trained using SGD with an initial learning rate 0.001 and which decays 10 times after 40k iterations, 0.9 momentum, 0.0005 weight decay, batch size of 64 (the ratio between positive and negative proposals is $1:1$), a total of 60k iterations.

Our framework is implemented using PyTorch [52] and runs on a workstation configured with an NVIDA P100 GPU card.

IV. EXPERIMENTS

A. Experiment Setup

We perform comprehensive studies on the challenging UA-DETRAC dataset [53] to demonstrate the outperformance of the proposed method, namely FG-BR Net. UA-DETRAC [53] contains 100 challenging video sequences corresponding to more than $140,000$ frames of real-world traffic scenes. There are more than 1.2 million vehicles labeled with bounding boxes in this dataset. The videos are recorded at 25 frames per seconds (fps), with the JPEG image resolution of 960×540 pixels. The evaluation metric of UA-DETRAC detection benchmark is strict: 0.7 IoU threshold is adopted for detecting cars. A website[1] is available for performance evaluation of object detection. The results of FR-BR Net, single model without any bells and whistles, has been made publicly available.

[1] http://detrac-db.rit.albany.edu/DetRet

For further detailed study of each component of FG-BR Net, a large-scale traffic surveillance dataset (TSD) is collected. It consists of videos from 18 different traffic scenes, with 13.2k frames for training and 11.8k for test. The test set can be further split into test set 1 (4.8k frames) from the same scenarios (but different videos) with the training set, and test set 2 (7k frames) from different scenes for testing generalization capability. The dataset has 5 object categories in total: car, bus, truck, cyclist and pedestrian. Both training and test sets contain scenarios in the daytime and nighttime, on overpass and intersection, with fixed cameras and cameras that horizontally rotate about 90 degrees every 20 minutes. We conduct experiments on TSD dataset to evaluate and analyze performance of the FG stage.

B. Experiments on TSD Dataset

The FG stage plays the role of Selective Search [60] in Fast RCNN [61], which selects high quality region proposals for the next stage. Moreover, the FG stage can be converted into a multi-scale single-shot detector by switching the classification task from class-agnostic to class-specific. The performance of FG as a detector reflects the quality of the region proposals when the it acts as a stage. Thus in this section, the FG stage is converted into a FG detector, to compare it with the state-of-the-art methods on TSD dataset.

As shown in TABLE I, the FG detector, with either ResNet18 or VGG16 backbone, outperforms the other models on both the-same-scenario (test set 1) and cross-scenario (test set 2) test sets in terms of mAP. In the following, we use FG-ResNet18 as our default FG detector and simply denote it by FG.

TABLE III shows FG is fairly good without up-sampling input frame sizes. The results show the FG outperforms SSD and YOLOv2 on small categories. MS-CNN performs better on small size objects, as its input size enlargement trick plays a critical role in improving its sensitivities on detecting small objects. However, MS-CNN models make very poor performance on the large category as the input enlargement makes it hard to detect too large objects out of their receptive fields after the enlargements. FG is more stable in detecting different sizes of objects.

In addition, the subsets with the scenes of nighttime and camera vibration are picked out from the test set to make them into a new test set, namely adverse scenarios in TABLE III. In these scenarios, the image quality is poor, which would lead to increment false positives of the detection algorithm. The results in TABLE III demonstrate that the FG is more robust than the other compared models on these challenging scenarios. FG performs the best among the compared models with at least 3-point lead. We suggest that feature-level masks, which are shown in Fig. 4, amplify feature activations on foreground objects and suppress the activations on background regions. Thus FG can reduce the number of false positives while increasing the recall rate.

Ablation Experiments on FG Stage. Ablation studies are performed to analyze the FG stage on TSD dataset. Results are reported in Table II and discussed in detail below.

Pixel-level mask is imperfect for object detection. We convert the foregrounds distilled by model F into binary pixel-level masks, and then use these masks directly for gating features in model A. Compared with model B that has no gating operation, A is worse almost in all terms. The results show that binary masks is not proper for object detection task as the masks are merely at pixel level.

Both original image and foreground are indispensable. Foreground images are directly involved in model C without feature gating block. Model C has the same structure with B except replacing the original images with foregrounds as inputs. C is worse than B in test set 1 while outperforming B in test set 2, which suggests foregrounds cannot completely replace original images. It is not surprising as useful contextual information could be removed during background subtraction process that may improve generalization capabilities of a detection model.

Masking operation is sensitive to its inputs. In model D, the original image rather than its foreground is used to generate feature-level masks, and the model E uses FG structure while keeping ρ in BS-RNN fixed to 0.9. Both model D and E perform better than no-masking model B, but model E outperforms model D. The results suggest that feature-level masking design is a more reasonable choice for the object detection task.

Fine-tuning element-wise $\{\rho_i\}_{i=1}^d$ improves detection. In model F we have element-wise $\{\rho_i\}_{i=1}^d$ and fine-tune it after it is initialized to 0.9 while training. The results show that model F is better than model E and performs the best among all these models. This suggests fine-tuning element-wise $\{\rho_i\}_{i=1}^d$ provides an extra flexibility to improve the detection performance.

In addition, we notice that several embedded smart camera based lightweight algorithms [62]–[65] introduced novel adaptive image processing techniques to achieve competitive performances with limited resources. In contrast, it is worth noting that the proposed BS-RNN block is deployed on NVIDIA GPU servers. In TSD dataset, BS-RNN processes videos at 0.6 milliseconds per frame with the resolution of 500×280. We suggest that although the whole framework (FG-BR Net) is designed for GPU servers, the proposed BS-RNN block is efficient for embedded systems.

C. Experiments on DETRAC Dataset

In this section experiments on UA-DETRAC [53] are conducted to compare with the state-of-the-art methods on this dataset. The FG-BR Net is trained on the whole training data, and evaluated on the test set by submitting the results to the official websites. It is worth noting that the FG-BR Net is trained as a single model without any bells and whistles, such as model/result ensemble, multi-scale testing and etc. TABLE V compares the proposed FG-BR Net with different methods evaluated on UA-DETRAC detection benchmark. The FG-BR Net ranks on the top among all the published methods on hard and sunny subsets. It is worth noting that the top two algorithms on the official ranking list, namely SSD_VDIG and HAVD, are not published as papers up to now.

TABLE V
DETECTION PERFORMANCE. AP SCORES OF DETECTION METHODS ON THE UA-DETRAC TEST SET IN VARIOUS ENVIRONMENTAL CONDITIONS. BOLD FACES ARE THE TOP PERFORMER ON EACH SUBSET.

method	overall	easy	medium	hard	cloudy	night	rainy	sunny
DPM [54]	25.70	34.42	30.29	17.62	24.78	30.91	25.55	31.77
ACF [55]	46.35	54.27	51.52	38.07	58.30	35.29	37.09	66.58
R-CNN [12]	48.95	59.31	54.06	39.47	59.73	39.32	39.06	67.52
CompACT [56]	53.23	64.84	58.70	43.16	63.23	46.37	44.21	71.16
GP-FRCNN [57]	76.57	91.79	80.85	66.05	81.23	77.20	68.59	85.16
EB [58]	67.99	87.77	73.03	54.74	75.13	71.80	52.99	82.04
SSDR [59]	59.07	77.84	64.41	45.98	62.79	60.88	48.55	74.32
FRCNN-Res [59]	61.65	82.90	66.89	48.14	61.97	65.88	59.13	59.17
DFCN [59]	65.82	86.83	72.96	50.47	69.90	69.41	54.11	80.79
HAVD	80.51	94.48	86.13	69.02	87.28	82.30	69.37	89.71
SSD_VDIG	**82.68**	**94.60**	**89.71**	70.65	**89.81**	**83.02**	**73.35**	88.11
FG-BR Net	79.96	93.49	83.60	**70.78**	87.36	78.42	70.50	**89.89**

Fig. 7. Explanation of how the image transformation operator deals with the camera vibrations. The left column shows the results of BS-RNN without transformation and the right column shows that of with transformation. The images of each column are original frame, foreground and background from top to bottom.

Therefore, the training and testing details of the two methods are not clear to the academic community. Apart from the two unpublished methods, the proposed FG-BR Net outperforms the other methods on detecting objects in surveillance videos. Fig. 8 shows some examples of the proposed method on UA-DETRAC test set, and Fig. 9 presents the precision-recall curves on the different UA-DETRAC test subset.

Ablation Experiments on BR Stage. Detailed analysis of the design of BR stage is reported below. Because the ground truth annotations of the UA-DETRAC testing set is not publicly available, we split the whole training set into mini-training set and validation set, 30 scenes in each set, containing runny, rainy, cloudy and night, to conduct analyses about our BR stage. The results on validation set are reported in TABLE IV and discussed in detail below.

The second stage helps detection. In baseline model, the FG stage is treated as a detector and outputs the detection results without background refining process. By contrast, model A switches the FG detector into the FG stage to supply region proposals for the second stage. It crops the proposals from original frames and uses ResNet18 as a backbone network without pairwise non-local operations. The results show that model A outperforms the baseline with about 3-point lead with both AP@0.5 and AP@0.7 metric. It indicates that the second stage refines the results based on the proposals form FG stage.

Normal non-local operation is good, and pairwise non-local is better. The normal non-local block [23] is added in model B. Model C replaces it with the proposed pairwise non-local operation. It crops the proposals from both original and background frames to compute the response at a position of original frames as a weighted average of the features at all positions of background frames. During training the BR stage, the batch size is set to 64 and the ratio of positive and negative samples to $1:1$. Specifically, we randomly choose 32 positive samples (IoU > 0.5) and 32 negative samples ($0.1 <$ IoU < 0.5) from all proposals in a frame, and the subsequent models (D and E) maintain the same sampling strategy. The result of B and C are better than A, which is just a normal R-CNN model. It is also worth noting that from model B to C, there is a greater improvement in AP@0.7 ($80.30\% \rightarrow 81.85\%$) than that in AP@0.5 ($85.93\% \rightarrow 86.31\%$). It demonstrates that

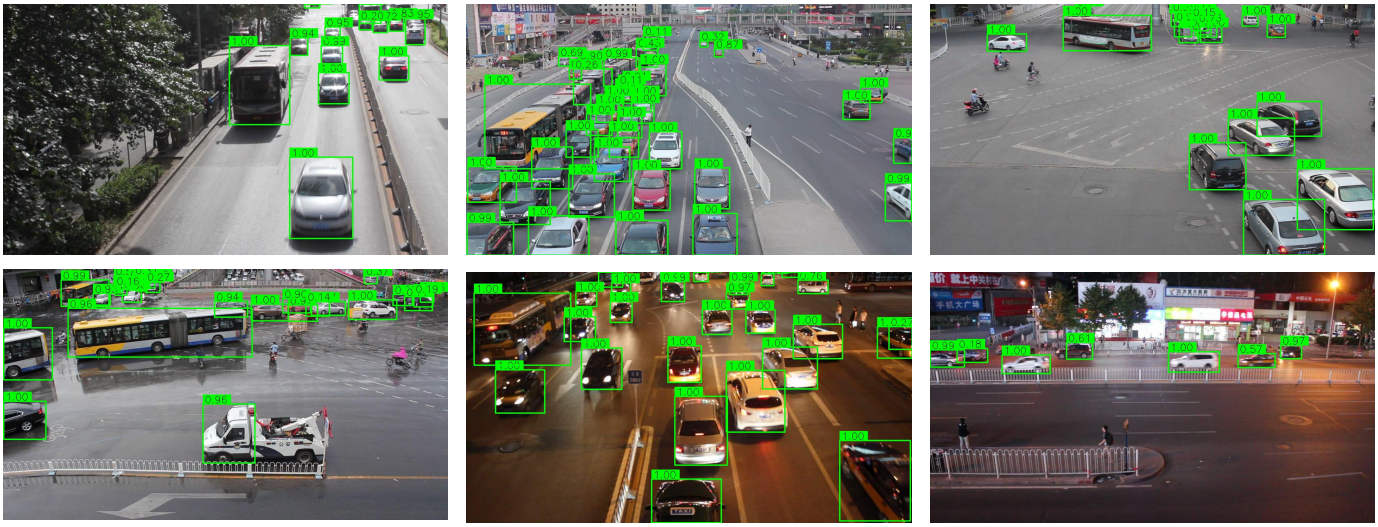


Fig. 8. Examples of detection results in various environmental conditional. A bounding box is plotted if its confidence score is larger than 0.1.

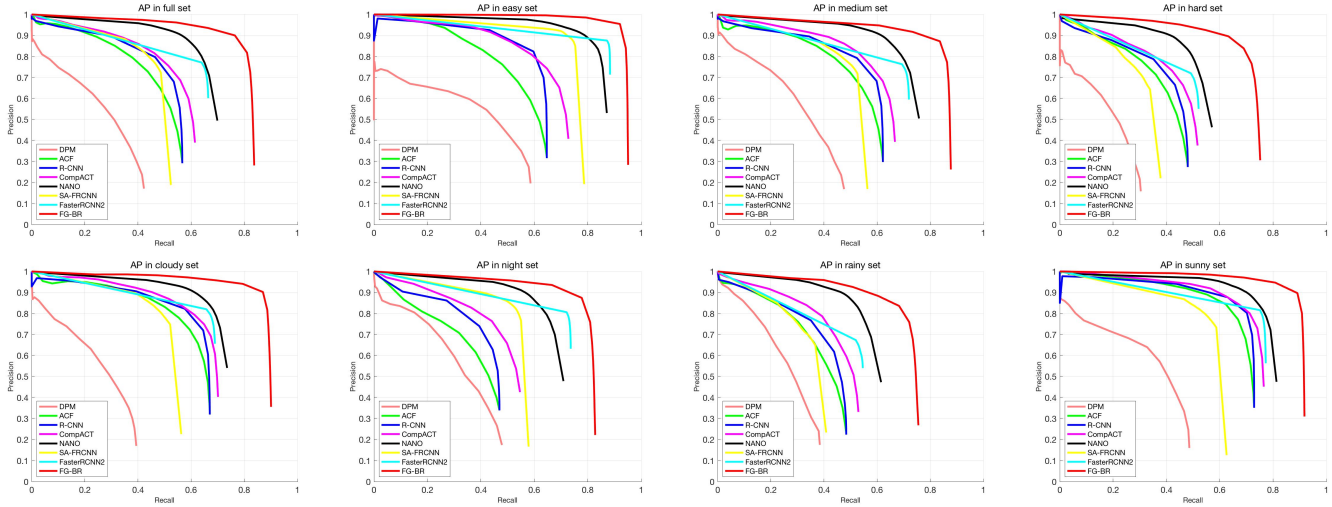


Fig. 9. Precision-recall curves on the UA-DETRAC test set of full, easy, medium, hard, cloudy, night, rainy and sunny respectively.

by pairwise non-local operations, the IoU of more detection results are regressed to higher values than 0.7.

More powerful backbone brings better performance. In model D, the backbone is replaced with ResNet50. It is not surprising that D outperforms C due to its more powerful backbone. And we also try different ratio of positive and negative samples in model E. The results suggest that the BR stage is not sensitive to the ratio of positive and negative samples with the comparison between model E and model D.

D. Analysis of Misalignment

Both qualitative and quantitative experiments are conducted to verify that the proposed method is effective in handling the misalignment problem.

First we add the image transformation operator τ in BS-RNN block. Fig. 7 illustrates how the operator τ improves the image quality. The camera vibrations lead to slight offsets in frames, and different offsets are applied on the background image frame by frame, causing the background to become blurred, which is shown in the left column of Fig. 7. In contrast, the right column shows sharper backgrounds and cleaner foregrounds by introducing the image transformation operator τ. A clean foreground image could reduce noises generated during the feature extraction in the FG stage, and a sharp background image could mitigate the misalignment problem together with pairwise non-local operation in the BR stage.

Next, the contributions of image transformation and pairwise non-local operation to the final detection performance are both investigated. We select all the subsets marked as "unstable" in the UA-DETRAC dataset, and divide them into training sets and validation sets, each containing 19 scenes. These scenes which are capture by unstable cameras, and it is suitable for investigating the effect of the method on misalignment problem. TABLE VI shows the results on this camera-unstable dataset. It shows that both image transformation and pairwise

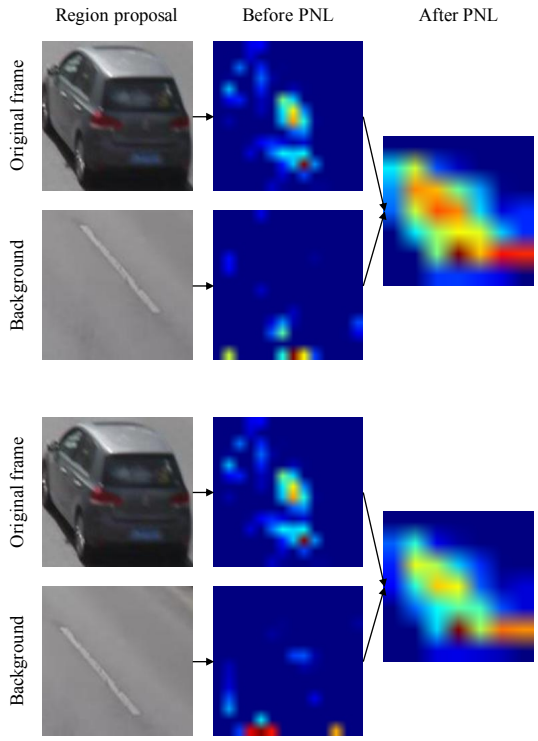


Fig. 10. Feature activations before and after pairwise non-local operation (PNL). For a better visualization, we select only one channel in the feature maps, then map their values to the interval of 0-255, and up-sample the feature channel by bilinear interpolation.

TABLE VI
RESULTS ON UNSTABLE CAMERA SETS.

model	AP@0.5	AP@0.7
FG w/o image trans.	78.88	72.63
FG w/ image trans.	80.12	73.36
FG w/ image trans. + BR	**82.56**	**74.35**

non-local operation are conductive to enhancing detection performance. Moreover, the BR stage does play the role of refining detection results to a high localization accuracy, and the two-stage method achieves the best results in TABLE VI. In addition, the features before and after pairwise non-local operations are visualized in Fig. 10. In this experiment the original region proposals are firstly used as an input pair to get the feature maps on the top row of Fig. 10. Then the background patch is manually shifted to the lower left corner by 20 pixels (in both the X and Y directions) to form a new input pair for the second row features. In Fig. 10, the features after pairwise non-local operations are similar with each other although the second input pair is misaligned with respect to the first pair.

V. CONCLUSION

This paper presented a two-stage object detection framework called Foreground Gating and Background Refining Network (FG-BR Net) for surveillance videos. In the foreground gating stage, the proposed method separates foregrounds and backgrounds, generates gating features to suppress the false positives. In the Background Refining stage, the proposed method introduces pairwise non-local operations to handle the misalignment problem. Extensive experiments showed that FG-BR Net outperforms the other state-of-the-art models on benchmark surveillance object detection datasets. In the future, we will study how to make the whole framework end-to-end for both training and inference.

ACKNOWLEDGMENT

This paper is partially supported by the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
[2] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *CVPR*, 2016.
[3] L. Zhang, T. Xiang, and S. Gong, "Learning a discriminative null space for person re-identification," in *CVPR*, 2016, pp. 1239–1248.
[4] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *CVPR*, 2014.
[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.
[6] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *ECCV*, 2014, pp. 345–360.
[7] G. Gkioxari, R. Girshick, P. Dollár, and K. He, "Detecting and recognizing human-object interactions," in *CVPR*, 2018.
[8] G. Gkioxari, B. Hariharan, R. Girshick, and J. Malik, "R-cnns for pose estimation and action detection," *arXiv preprint arXiv:1406.5212*, 2014.
[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.
[10] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *CVPR*, 2017.
[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.
[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.
[13] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *ECCV*, 2016.
[14] Z. Fu, Z. Jin, G.-J. Qi, C. Shen, R. Jiang, Y. Chen, and X.-S. Hua, "Previewer for multi-scale object detector," in *2018 ACM Multimedia Conference on Multimedia Conference (MM)*. ACM, 2018, pp. 265–273.
[15] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *arXiv preprint arXiv:1703.06211*, 2017.
[16] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.
[17] C. Zhan, X. Duan, S. Xu, Z. Song, and M. Luo, "An improved moving object detection algorithm based on frame difference and edge detection," in *ICIG*, 2007, pp. 519–523.
[18] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*, 1999.
[19] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *CVPR*, 2012.
[20] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.
[21] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *arXiv preprint arXiv:1511.04136*, 2015.
[22] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.

[23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," *CVPR*, 2018.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008.

[25] Z. Fu, C. Zhou, H. Yong, R. Jiang, X. Tian, Y. Chen, and X.-S. Hua, "Foreground gated network for surveillance object detection," in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE, 2018, pp. 1–7.

[26] Y. Khandhediya, K. Sav, and V. Gajjar, "Human detection for night surveillance using adaptive background subtracted image," *arXiv preprint arXiv:1709.09389*, 2017.

[27] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia, "Noscope: optimizing neural network queries over video at scale," *Proceedings of the VLDB Endowment*, 2017.

[28] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer science review*, vol. 11, pp. 31–66, 2014.

[29] L. Maddalena and A. Petrosino, "Background subtraction for moving object detection in rgbd data: A survey," *Journal of Imaging*, vol. 4, no. 5, p. 71, 2018.

[30] T. Bouwmans and B. Garcia-Garcia, "Background subtraction in real applications: Challenges, current models and future directions," *arXiv preprint arXiv:1901.03577*, 2019.

[31] N. J. McFarlane and C. P. Schofield, "Segmentation and tracking of piglets in images," *Machine vision and applications*, vol. 8, no. 3, pp. 187–193, 1995.

[32] B. Lee and M. Hedley, "Background estimation for video surveillance," 2002.

[33] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust pca, robust subspace tracking, and robust subspace recovery," *IEEE signal processing magazine*, vol. 35, no. 4, pp. 32–55, 2018.

[34] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust pca in image and video processing," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1427–1457, 2018.

[35] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2016.

[36] H. Guo, C. Qiu, and N. Vaswani, "Practical reprocs for separating sparse and low-dimensional signal sequences from their sum part 1," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4161–4165.

[37] H. Guo, N. Vaswani, and C. Qiu, "Practical reprocs for separating sparse and low-dimensional signal sequences from their sum part 2," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2014, pp. 369–373.

[38] P. Narayanamurthy and N. Vaswani, "A fast and memory-efficient algorithm for robust pca (merop)," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4684–4688.

[39] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *ICASSP*, 2008.

[40] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *CVPR*, vol. 2. IEEE, 2005, pp. 60–65.

[41] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *ICCV*. IEEE, 2009, pp. 349–356.

[42] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *ICCV*. IEEE, 2009, pp. 2272–2279.

[43] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," in *CVPR*, 2015, pp. 3992–4000.

[44] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *CVPR*, 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[46] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017.

[47] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "Ron: Reverse connection with objectness prior networks for object detection," in *CVPR*, 2017.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[49] M. J. Shafiee, P. Siva, P. Fieguth, and A. Wong, "Embedded motion detection via neural response mixture background modeling," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2016, pp. 837–844.

[50] J. Dou, Q. Qin, and Z. Tu, "Background subtraction based on deep convolutional neural networks features," *Multimedia Tools and Applications*, pp. 1–23, 2018.

[51] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, "Deep neural network concepts for background subtraction: A systematic review and comparative evaluation," *arXiv preprint arXiv:1811.05255*, 2018.

[52] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[53] L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang, and S. Lyu, "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking," *arXiv CoRR*, vol. abs/1511.04136, 2015.

[54] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[55] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[56] Z. Cai, M. Saberian, and N. Vasconcelos, "Learning complexity-aware cascades for deep pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3361–3369.

[57] S. Amin and F. Galasso, "Geometric proposals for faster r-cnn," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[58] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," in *Multimedia and Expo (ICME), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1135–1140.

[59] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco *et al.*, "Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[60] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

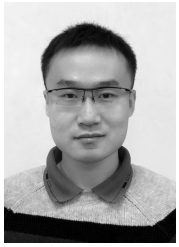
[61] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[62] M. Casares, S. Velipasalar, and A. Pinto, "Light-weight salient foreground detection for embedded smart cameras," *Computer Vision and Image Understanding*, vol. 114, no. 11, pp. 1223–1237, 2010.

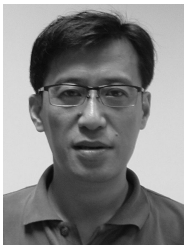
[63] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1438–1452, 2011.

[64] S. Apewokin, B. Valentine, J. Choi, L. Wills, and S. Wills, "Real-time adaptive background modeling for multicore embedded systems," *Journal of Signal Processing Systems*, vol. 62, no. 1, pp. 65–76, 2011.

[65] Y. Wang, S. Velipasalar, and M. Casares, "Cooperative object tracking and composite event detection with wireless embedded smart cameras," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2614–2633, 2010.



Zhihang Fu received his B.S. degree from Zhejiang University, Hangzhou, China, in 2013. He will receive his Ph.D. degree from Zhejiang University, Hangzhou, China in July, 2019. He is currently a research intern at Alibaba DAMO Academy, Hangzhou, China. His research interests include deep learning and computer vision.



Yaowu Chen was born in Heilongjiang Province, China, in 1963. He received the Ph.D. degree from Zhejiang University, Hangzhou, China, in 1998. He is currently a professor and the director of the Institute of Advanced Digital Technologies and Instrumentation, Zhejiang University. His major research fields are embedded systems, multimedia systems, and networking.

Hongwei Yong received the B.Sc. and M.Sc. degrees from Xian Jiaotong University, Xian, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His current research interests include image modeling and deep learning.

Rongxin Jiang received his B.S. and Ph.D. degrees in Electrical Engineering from the College of Biomedical Engineering & Instrument Science, Zhejiang University, China, in 2002 and 2008 respectively. He is currently an associate researcher at the College of Biomedical Engineering & Instrument Science, Zhejiang University, China. His research interests include embedded system and computer vision.

Lei Zhang (M'04-SM'14-F'18) received his B.Sc., M.Sc. and Ph.D degrees in 1995, 1998 and 2001, respectively. Before he joined Alibaba DAMO Academy in Sept. 2018, he was a Chair Professor in Dept. of Computing, The Hong Kong Polytechnic University. Prof. Zhang was an IEEE Fellow for his contributions in sparsity based image modeling and image quality assessment. His research interests include Computer Vision, Image and Video Analysis, Pattern Recognition, and Biometrics, etc. Prof. Zhang has published more than 200 papers in those areas. His publications have been cited more than 40,000 times in literature. He was selected as a "Clarivate Analytics Highly Cited Researcher" consecutively from 2015 to 2018. Prof. Zhang is/was an Associate Editor of IEEE Trans. on Image Processing, SIAM Journal of Imaging Sciences, IEEE Trans. on CSVT, and Image and Vision Computing, etc.

Xian-Sheng Hua (M'05-SM'14-F'16) received the B.S. and Ph.D. degrees in applied mathematics from Peking University, Beijing, China, in 1996 and 2001, respectively. He joined Microsoft Research Asia, Beijing, China, in 2001, as a Researcher. He was a Principal Research and a Development Lead in multimedia search with the Microsoft Search Engine, Bing, Redmond, WA, USA, from 2011 to 2013. He was a Senior Researcher with Microsoft Research Redmond, Redmond, from 2013 to 2015. He became a Researcher and the Senior Director of the Alibaba Group, Hangzhou, China, in 2015, where he is also leading the Visual Computing Team, Search Division, Alibaba Cloud, and then DAMO Academy. He is currently a Distinguished Engineer/Vice President of the Alibaba Group, where he is leading a team working on large-scale visual intelligence on the cloud. He has authored or co-authored more than 200 research papers and has filed more than 90 patents. His research interests include big multimedia data search, advertising, understanding, and mining, pattern recognition, and machine learning. He is an IEEE Fellow and an ACM Distinguished Scientist. He was one of the recipients of the 2008 MIT Technology Review TR35 Young Innovator Award for his outstanding contributions on video search. He was also a recipient of the Best Paper Awards at ACM Multimedia 2007, and the Best Paper Award of the IEEE Transactions on Circuits and Systems for Video Technology in 2014. He served as a Program Co-Chair for IEEE ICME 2012, ACM Multimedia 2012, and IEEE ICME 2013. He will be serving as a General Co-Chair of ACM Multimedia in 2020.