IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 10, 2020, accepted November 15, 2020, date of publication November 18, 2020,
date of current version December 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038913

Video Object Detection Guided by Object Blur Evaluation

YUJIE WU, HONG ZHANG, YAWEI LI, YIFAN YANG, AND DING YUAN, (Member, IEEE)
Image Processing Center, Beihang University, Beijing 100191, China

Corresponding author: Ding Yuan (dyuan@buaa.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872019, and in part by the National Key Research and Development Program of China under Grant 2016YFE0108100.

ABSTRACT In recent years, the excellent image-based object detection algorithms are transferred to the video object detection directly. These frame-by-frame processing methods are suboptimal owing to the degenerate object appearance such as motion blur, defocus and rare poses. The existing works for video object detection mostly focus on the feature aggregation at pixel level and instance level, but the blur impact in the aggregation process has not been exploited well so far. In this article, we propose an end-to-end blur-aid feature aggregation network (BFAN) for video object detection. The proposed BFAN focuses on the aggregation process influenced by the blur including motion blur and defocus with high accuracy and little increased computation. In BFAN, we evaluate the object blur degree of each frame as the weight for aggregation. Noteworthy, the background is usually flat which has a negative impact on the object blur degree evaluation. Therefore, we introduce a light saliency detection network to alleviate the background interference. The experiments conducted on the ImageNet VID dataset show that BFAN achieves the state-of-the-art detection performance, exactly 79.1% mAP, with 3 points improvement compared to the video object detection baseline.

INDEX TERMS Video object detection, object blur degree evaluation, saliency detection.

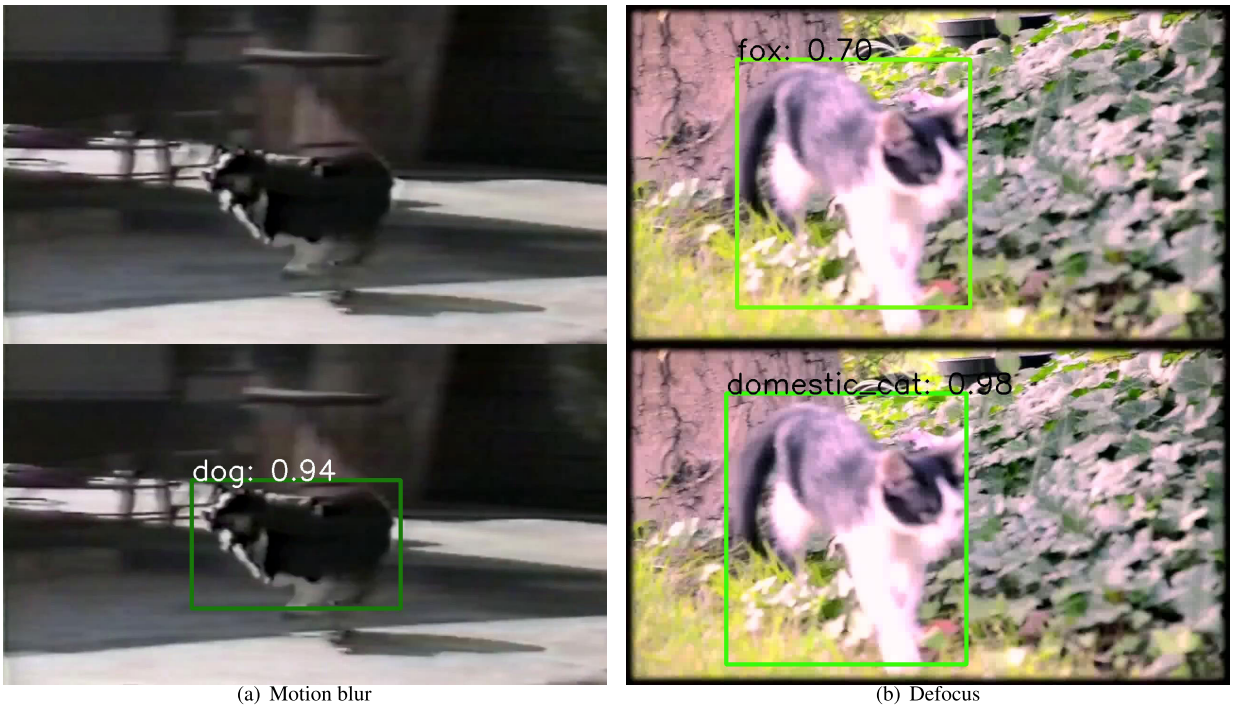
I. INTRODUCTION

The deep learning network has achieved significant progress in object detection [1], [2]. Compared to the still image object detection, the video object detection is more challenging because the drastic appearance variation occurs in the video frames. The common appearance variation is blur, which could decrease the detection accuracy to a great extent. As shown in Figure 1, the motion blur and the defocus both discourage the accurate inference of the still image detector [3]. Therefore, the exploitation of the blur information is beneficial to video object detection.

The state-of-the-art video object detection algorithms can be categorized into two types: box-level post processing and feature aggregation. In the early stage, the box-level post-processing methods combined the CNN based still image detector and the tracker on the detected bounding box [4]. This kind of methods first applied the still image object detector, and then manipulated the detected bounding box across the temporal dimension as a dedicated post processing step. T-CNN [5] and D&T [6] both improved the detection accuracy on the basis of the still image object detector by optimizing the detected bounding box. Moreover, these methods were not trained end-to-end since the generation of the proposal boxes and the box-level post processing are independent. Although these methods achieved promising results compared to the still image detectors, they were computationally expensive. To tackle the misalignment among the ajacent frames [7], the other solution based on feature aggregation become the mainstream to this end. These methods constructed the connections among frames on the feature level including pixel and instance levels. The feature aggregation based methods achieved higher accuracy with higher efficiency compared to the former box-level post processing methods owing to end-to-end training. The proposed method belongs to the latter elegant framework.

The existing feature aggregation based methods compensate the misalignment among frames by aggregating features of many ajacent frames. One critical issue is that whether these frame should be treated equally. There are two existing

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.



(a) Motion blur (b) Defocus

FIGURE 1. The detection results from Faster R-CNN (the top row) and BFAN (the bottom row) on two blur frames. (a) and (b) are the frames suffering from motion blur and defocus, respectively. The Faster R-CNN fails to produce a prediction for the motion blur case, and it also gives a wrong prediction for the defocus case. The proposed method succeeds in the inference with a high confidence in comparison.

solutions to answer the issue. One solution is to treat each frame equally and assign them the same weight. The other one is to adopt a light network to learn the weight in the training process. These two solutions both lack of the special consideration for the blur influence. The blur influence has been considered to discriminatively treat consecutive frames for saliency detection [8]. To address the limitation of the existing methods, we focus on the blur influence in the feature aggregation process and develop a blur-aid feature aggregation network (BFAN) for video object detection. In our opinion, because the appearance of the objects in some frames deteriorates due to the motion blur or defocus, these frames are supposed to be assigned low weights. On the contrary, the frames whose object appearance is clear are supposed to be assigned high weights. Specifically, we evaluate the object blur degree of each frame and use it as the weight of the frame. In this way, the frames whose object appearance is clear make more contribution to the result than those whose object appearance is blur. Moreover, we are only concerned about the blur degree of the objects, not the whole frame. The background is usually sophiscated, which could disturb the object blur degree evaluation. A light saliency detection network is introduced to alleviate the negative impact of the background [9]. We thereby only evaluate the blur degree of the objects to be detected in each frame.

The main contribution of this article can be summarized as follows:

- We advocate a novel BFAN focusing on the blur influence in video object detection. The frames whose object appearance is clear contribute more to the result than those frames whose object appearance is blur. In this way, the blur objects are easier to be detected guided by other clear frames.
- Because we only care the blur degree of the objects, the background interference could decrease the detection accuracy. We adopt a light saliency detection network to alleviate the background interference.
- The experiments on VID dataset exhibit that the proposed method achieves the state-of-the-art performance, exactly 79.1% mAP, which increases by 3% compared to the adaptive weight video object detection baseline.

The remaining part of this article has been organised in the following way. Section II gives a brief review of the existing object detection algorithms and blur estimation methods. Section III describes our implementation in detail. The experimental results and the ablation analysis are shown in Section IV. Finally, the conclusion is provided in Section V.

II. RELATED WORK

In this section, we first review the still image and video object detection algorithms. Then a short review of the blur mapping methods is given afterwards.

A. STILL IMAGE OBJECT DETECTION

We have witnessed the great success of CNN in various domains such as image classification [10], object detection [11] and image restoration [12]. The object detection for still images is one of the most successful domains until

now. R-CNN [13] first used the CNN to extract the features followed by SVM classification, and it improved the accuracy greatly compared to the traditional detection methods such as DPM [14]. The following Fast R-CNN [15] and Faster R-CNN [3] improved the ROI pooling and put forward the region proposed network to accelerate speed and improved accuracy. The above three detection algorithms belong to the two-stage framework. They give the proposal bounding box, and then classify the conrresponding features. Different from the two-stage framework, YOLO [16] and SSD [17] are the representatives of the one-stage framework. They predict the categories and the locations without the proposal bounding boxes, which is more efficient.

B. VIDEO OBJECT DETECTION

Compared to the datasets for still image object detection such as PASCAL VOC [18] and COCO [19], the first dataset for video object detection ImageNet VID [20] was introduced in 2015. The early algorithms paid attention to the box-level post processing. T-CNN [5] and D&T [6] introduced the tracking on the detected bounding boxes. Because the object detection on each frame and the post tracking are independent, these methods were difficult to be trained end-to-end and they were computative expensively. Afterwards, Zhu *et al.* put forward the first two feature aggregation methods DFF [21] and FGFA [22]. These two methods dug into detection network and they were more efficient because of the end-to-end training. DFF and FGFA both introduced the flow guided warping to optimize the detection process, but they pursued the high efficiency and the high accuracy, respectively. DFF improved the feature extraction process, and it utilized the flow to get the feature of the non-key frame based on the key frame, which saved much time. However, FGFA focused on the accuracy improvement by aggregating the features of ajacent frames via flow. The subsequent methods MANet [23] and MFCN [24] took the instance level feature into consideration to boost the performance. STMM [25] and LWDN [26] adopted the memory network such as LTSM to balance the accuracy and speed. STSN [27] and SSVD [28] considered the sampling stream other than the motion stream via deformable convolution [29]. The above video object detection algorithms exploited the motion stream and sampling stream on pixel level and instance level for the compensation. However, the special design for blur influence is still blank until now.

C. BLUR MAPPING

One important step of our algorithm is to evaluate the blur degree of the objects, not the whole frame. Therefore the existing works about the blur degree for the whole frame are not suitable for our work. We utilize the blur mapping method to label every pixel as either blurry or non-blurry, and then extract the object parts guided by the saliency detection. The first representative dataset for blur mapping was proposed by Shi *et al.* [30], and it contained two types of blur: motion blur and out-of-focus. The subsequent blur mapping methods focused on either the out-of-focus [31] or the motion blur [32]. The most existing blur mapping algorithms were based on hand-craft features, thus they were not robust enough to discriminate the truly blur region and the flat region in the nature such as the sky. Futhermore, the proposed method in this article is based on CNN, hence a blur mapping method based on CNN is essential due to the end-to-end training. Ma *et al.* proposed a deep blur mapper (DBM) [33] to separate the truly blur region including motion blur and out-of-focus from the whole image robustly. Moreover, DBM was a fully convolutional network which could be utilized as one part of our whole network in the end-to-end training process.

III. PROPOSED METHOD

A. OVERVIEW

As shown in Figure 2, the overall architecture of BFAN is based on the pixel level feature aggregation framework [22]. Our model contains a feature extraction backbone N_{feat} that generates the deep intermediate feature, a flow network N_{flow} that calibrates the features from the supporting frames such as $f_{t-\tau}$ and $f_{t+\tau}$, a module that produces the weights for frames as shown in Figure 2 (b) and a detection head N_{det} that gives the final detection results including categories and locations. The input frames are partitioned into the reference frame I_t and the supporting frames $I_{t-\tau}, I_{t+\tau}$. The supporting frames provide the information to boost the detection performance for the reference frame.

Our main contribution is a feature aggregation based detection framework guided by the object blur evaluation. We illustrate the proposed model in three stages step by step: 1. the pixel level feature aggregation for detection; 2. the weight calculation guided by the object blur evalution; 3. the object blur evaluation calibrated by the saliency detection. In the following section, we describe above three parts in detail.

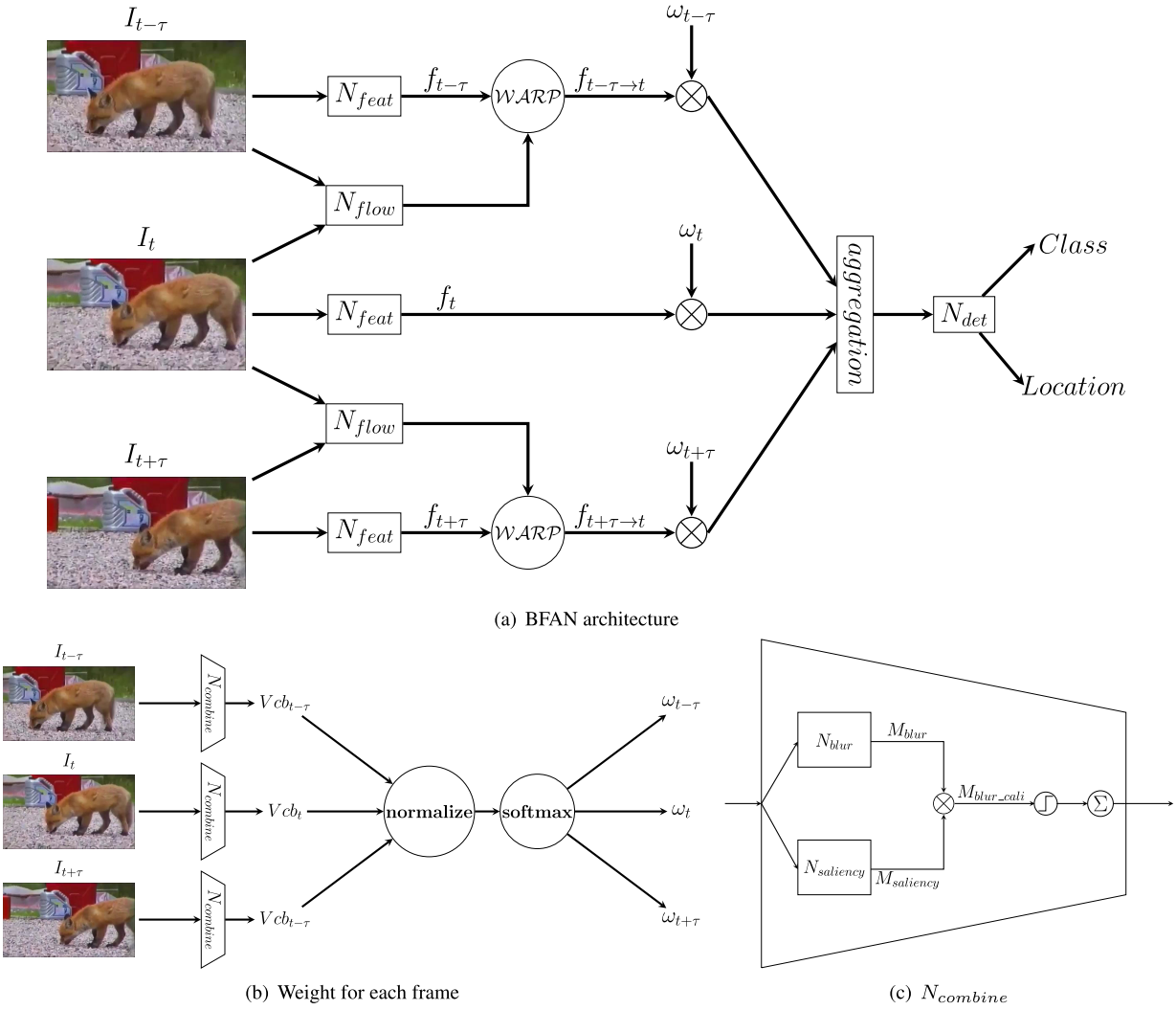
1) PIXEL LEVEL AGGREGATION

It is a consensus that there are movement among frames in a video. To efficiently utilize the information of the ajacent frames, a flow network [34] is recommended to compensate the misalignment due to the movement. Given a reference frame I_t and a supporting frame $I_{t-\tau}$, the flow network N_{flow} can estimate the flow field $\mathcal{M}_{t-\tau\to t} = N_{flow}(I_{t-\tau}, I_t)$. The flow field $\mathcal{M}_{t-\tau\to t}$ predicts the distance from the pixel in $I_{t-\tau\to t}$ to the corresponding pixel in I_t. Therefore, the feature of the supporting frame is warped to the reference frame according to the flow field as follows:

$$f_{t-\tau\to t} = \mathcal{W}(f_{t-\tau}, \mathcal{M}_{t-\tau\to t}), \tag{1}$$

where $f_{t-\tau\to t}$ is the warped feature from frame $I_{t-\tau}$ to I_t and $\mathcal{W}(\cdot)$ denotes the bilinear warping function.

With the warped features of the supporting frames, we have accumulated the information from nearby frames for the reference frame. These features from ajacent frames provide much useful information to make up for the weakness of



(a) BFAN architecture

(b) Weight for each frame

(c) $N_{combine}$

FIGURE 2. The proposed BFAN architecture. We take three frames $t-\tau, t, t+\tau$ as an example for illustration. (a) is the whole architecture. Each frame is fed into the feature extraction network N_{feat} to obtain its own deep convolutional feature $f_{t-\tau}, f_t, f_{t+\tau}$. The flow network N_{flow} is utilized to get the flow map between two frames, which is used to compensate the motion misalignment by $\mathcal{WARP}$, namely $f_{t-\tau\to t}, f_{t+\tau\to t}$. The warped features $f_{t-\tau\to t}, f_{t+\tau\to t}$ and the feature of the reference frame f_t are assigned with different weights $\omega_{t-\tau}, \omega_{t+\tau}, \omega_t$, respectively. (b) generates the weights $\omega_{t-\tau}, \omega_t, \omega_{t+\tau}$ for frames $I_{t-\tau}, I_t, I_{t+\tau}$, respectively. Each frame is fed into the blur mapping network N_{blur} and the saliency detection $N_{saliency}$ network simultaneously as shown in (c). The blur map M_{blur} is dot multiplied ($\otimes$) by the saliency map $M_{saliency}$ to obtain the calibrated blur map M_{blur_cali}. Then a step function with threshold 0.5 is utilized for binarization. The sum of whole blur map binarization is used as the calibrated blur value *Vcb* of the frame. Lastly, all calibrated blur values are normalized and mapped into [0,1] by softmax function to achieve weigths $\omega_{t-\tau}, \omega_t, \omega_{t+\tau}$.

the reference frame such as rare poses and blur appearance. For aggregation, there are two common solutions. One solution is to assign each feature with the same weight, i.e. treating all features equally. The other solution is to assign each feature with different weight. One representative of the second solution is to adopt a tiny network to predict the weight for each frame, and the parameters of the tiny network are optimized in the training process. Different from the adaptive weight, we introduce the object blur evaluation to guide the weight, which is illustrated in next subsection. Finally, the aggregated feature is fed into to detection network N_{det} to produce categories and locations for objects.

2) WEIGHT GUIDED BY THE OBJECT BLUR EVALUATION

As shown in Figure 2 (b), each frame is fed into the combine network $N_{combine}$ ($N_{combine}$ is described in Figure 2 (c) in detail) to obtain the value, which stands for the blur degree of the frame. Because the values are too large to mapped into [0,1] by softmax function, we first normalize all values as follows:

$$VcbNorm_i = \frac{Vcb_i}{\sqrt{{Vcb_{t-\tau}}^2 + {Vcb_t}^2 + {Vcb_{t+\tau}}^2}}, \quad i \in \{t-\tau, t, t+\tau\}, \tag{2}$$

where *Vcb* denotes the calibrated blur value and *VcbNorm* is the normalization result for *Vcb*. Followed by the softmax

function, *VcbNorm* is converted to the weight ω for each frame.

3) OBJECT BLUR EVALUATION CALIBRATED BY THE SALIENCY DETECTION

The details of $N_{combine}$ is shown in Figure 2 (c). Each frame is fed into the the blur map network N_{blur} and the saliency detection network $N_{saliency}$ simultaneously. The blur mapping network N_{blur} is able to label each pixel as either blur or non-blur. However, we only care about the blur degree of the objects, thus the background interference is supposed to be excluded. Therefore, a saliency network $N_{saliency}$ is adopted to extract the region of interest, which could alleviate the background interference to a great extent. The blur map is calibrated by alleviating the background interference via dot multiplication with the saliency map.

With the calibrated blur map M_{blur_cali}, we utilize a step function for binarization as follows:

$$u(t) = \begin{cases} 1, & t > 0.5, \\ 0, & otherwise. \end{cases} \tag{3}$$

Finally, all pixels are accumulated in the whole map to achieve *Vcb* in Figure 2 (b).

B. MODEL ARCHITECTURE

The proposed BFAN contains five essential subnetworks: the feature extraction network, the flow estimation network, the blur mapping network, the saliency detection network and the detection network. Firstly, the feature extraction network extract the deep feature from the input frames. Secondly, the flow estimation network estimated the flow field between two arbitrary frames to obtain the warped features. Thirdly, the blur mapping network and the saliency network extract the blur map and the saliency map, respectively. The saliency map is utilized to alleviate the background interference for object blur evaluation, i.e. the weights for frames. Finally, the warped features multiplied by the corresponding weights are aggregated to fed into the detection network, and the objects of interest are obtained. To design these five subnetworks is out of scope of this article, and there are many existing works focusing on each special field. We hence employ the existing networks directly, and describe them below.

1) FEATURE EXTRACTION NETWORK

We choose the Resnet-101 [35] as the feature extraction network. In order to extract the feature for subsequent process, we remove the last average pooling and fully-connected layers. Following the same strategy in [22], we enlarge the resolution of the feature maps by changing the stride of the first convolutional layers in the conv5 from 2 to 1. Furthermore, the dilation of these convolutional layers is set as 2 to keep the receptive field.

2) FLOW ESTIMATION NETWORK

There are many existing work focusing on flow estimation such as FlowNet [34], FlowNet2 [36], PWC-Net [37]. Since the state-of-the-art video object detection algorithms mostly use the Flownet (the simple version), we follow the same strategy for fairness. As there is a mismatch between the resolution of the output flow field and the resolution of the feature maps from the feature extraction network, we resize the flow field to match the feature maps.

3) BLUR MAPPING NETWORK

Our goal is to evaluate the blur degree of the object, and the blur includes motion blur and out-of-focus. The existing blur mapping algorithms are mostly designed for either motion blur or out-of-focus, which cannot meet our demand. We choose the Deep blur mapping (DBM) [33] as blur mapping network. Because DBM is able to discriminate the motion blur and out-of-focus at the same time, and it is robust enough to distinguish the out-of-focus and flat region. Furthermore, DBM is an end-to-end fully convolutional network, which is convenient for training. Therefore, DBM is the best choice for the proposed model.

4) SALIENCY DETECTION NETWORK

To alleviate the impact of the background, we introduce a saliency detection network. Most saliency detection networks are computative expensively [38]–[40], thus they are unsuitable for the proposed method. We choose a light saliency detection network CSNet [41]. CSNet reduces the representative redundancy with a flexible convolutional module, i.e. gOctConv, and it achieves comparable performance with only 0.2% parameters. The experimental results show that CSNet improves the detection performance of BFAN, and the increased computation is very little.

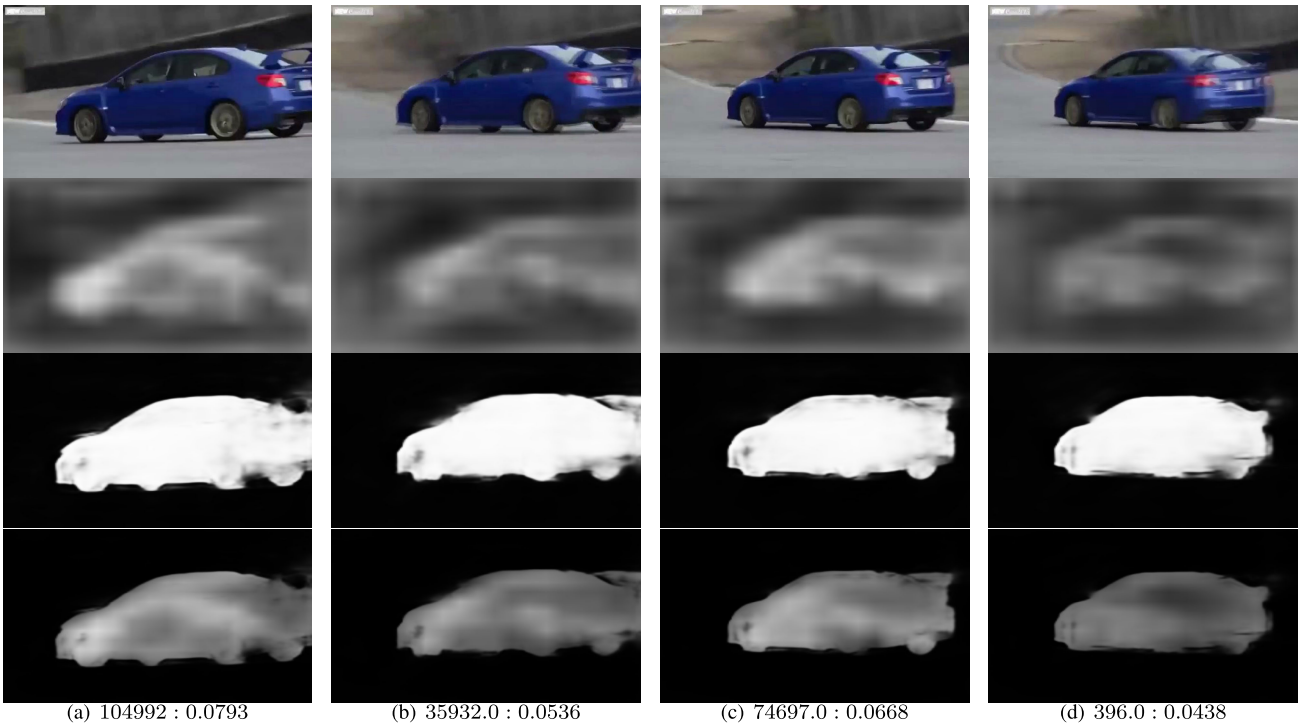
As shown in Figure 3, the orginal images and their corresponding blur maps, saliency maps and calibrated blur maps are listed from the top row to the bottom row. The cars in (b) and (d) both contain motion blur compared to the cars in (a) and (c). The blur maps in (b) and (d) are darker than those in (a) and (c). The saliency maps in the third row are able to alleviate the background interference, thus the calibrated blur maps which only care about the object blur degree are obtained in the bottom row. Therefore, the frames in (a) and (c) are assigned higher weights in feature aggregation.

5) DETECTION NETWORK

We mainly use the Faster R-CNN [3] as our default detection network. Different the orginal setting in Faster R-CNN, we choose 12 anchors for each position in Region Proposal Network (RPN). The 12 anchors includes 3 aspect ratios {1:2, 1:1, 2:1} and 4 scales $\{64^2, 128^2, 256^2, 512^2\}$. We choose 300 anchors for each frame with an NMS threshold 0.7 in the training and inference process. Finally, the ROI-Align layer followed by a 1024-D fully-connected layer after conv5 stage is utilized for classification.

C. MODEL INFERENCE

Algorithm 1 shows the inference procedure of BFAN in detail. Given the input video frames $\{I_i\}$ and the aggregation



(a) 104992 : 0.0793 (b) 35932.0 : 0.0536 (c) 74697.0 : 0.0668 (d) 396.0 : 0.0438

FIGURE 3. The visual effect of the blur mapping network and the saliency detection network. From top to bottom are the original image, the blur map, the saliency map and the calibrated blur map, respectively. The labels under the images are *Vcb* : ω as shown in Figure 2. The car in (a) and (c) are clearer then car in (b) and (d), thus the former two frames are assigned higher weights. Best viewed in pdf, zoom-in.

range K, BFAN sequentially processes the frames with a $2K + 1$ range as the supporting frame set. We construct a butter to store the feature maps and the value of calibrated blur map (Vcb) of each frame in the supporing frame set. However, at the begin K frames and the end K frames, we replicate the first frame and the last frame to fill the butter, respectively. At the beginning, we extrace the feature maps and the values of calibrated blur map (Vcb) of the first $K + 1$ frames to initialize the butter (L2-L6 in Algorithm 1). Moreover, we replicate the feature maps and Vcb of the first frame K times to make the buffer contain $2K + 1$ frames (L8-L9 in Algorithm 1). With the initialized buffer, BFAN sequentially processes the video frames (L11-L17 in Algorithm 1) and update the buffer (L18-L23 in Algorithm 1). For the i-th reference frame, the feature maps of the supporting frames are warped to the reference frame (L12 in Algorithm 1). The warped features are aggregated with the corresponding weights (L14-L16 in Algorithm 1). Finally, the aggregated feature is fed into the detection network to obtain categories and locations of the objects (L17 in Algorithm 1).

D. COMPLEXITY ANALYSIS

According to Algorithm 1, we analyze the complexity of BFAN. Aside from the feature extraction network N_{feat}, BFAN contains following modules: the flow estimation network N_{flow}, the warp bilinear function $\mathcal{W}$, the blur mapping network N_{blur}, the saliency detection network $N_{saliency}$, the weight calculation denoted as ϵ (dot multiplication, step function, accumulation, normalization and softmax) and the detection network N_{det}. For the supporting frames range K, the complexity of the proposed method is

$$\begin{aligned}\mathcal{O} = \mathcal{O}(N_{feat}) + \mathcal{O}(N_{det}) + \mathcal{O}(N_{blur}) + \mathcal{O}(N_{saliency}) \\ + (2K+1)(\mathcal{O}(N_{flow}) + \mathcal{O}(\mathcal{W}) + \mathcal{O}(\epsilon)),\end{aligned} \tag{4}$$

where $\mathcal{O}$ measures the complexity. Compared to the still image detector, the ratio of BFAN versus Faster R-CNN is

$$\begin{aligned}r = 1 + \{\mathcal{O}(N_{blur}) + \mathcal{O}(N_{saliency}) + (2K+1)(\mathcal{O}(N_{flow}) \\ + \mathcal{O}(\mathcal{W}) + \mathcal{O}(\epsilon))\}/\{\mathcal{O}(N_{feat}) + \mathcal{O}(N_{det})\},\end{aligned} \tag{5}$$

Typically, the complexity of N_{det}, $\mathcal{W}$, ϵ can be ignored compared to N_{feat}. The ratio hence is approximated as follows:

$$r \approx 1 + \frac{\mathcal{O}(N_{blur}) + \mathcal{O}(N_{saliency}) + (2K+1)\mathcal{O}(N_{flow})}{\mathcal{O}(N_{feat})}. \tag{6}$$

Therefore, the increased computational cost mainly comes from the blur mapping network, the saliency detection network and the flow estimation network. The blur mapping network and the flow estimation network are both fully connected network, and they are of nearly the same complexity. The complexity of these two network is much lower than N_{feat} in general [22]. As for $N_{saliency}$, it is a very light network whose complexity is much more lower than N_{feat}. As shown in the following execution time Table 2, the increased computational time is affordable.

Algorithm 1 Local Search Based Algorithm

Input: video frames $\{I_i\}$, aggregation range K
Output: detection results y_i

for $k = 1; k++; k \leq K+1$ **do**
  $f_k = N_{feat}(I_k)$ ▷ Initialize feature and weight butter
  $M_{blur\_k} = N_{blur}(I_k)$ ▷ Calculate the blur map
  $M_{saliency\_k} = N_{saliency}(I_k)$ ▷ Calculate the saliency map
  $M_{blur\_cali\_k} = M_{blur} \otimes M_{saliency}$ ▷ Calibrate the blur map with the saliency map
  $Vcb_k = \Sigma u(M_{blur\_cali\_k})$
end for
$f_{set} = \{\underbrace{f_1, f_1, \ldots, f_1}_{K}, f_1, f_2, \ldots, f_{K+1}\}$ ▷ Construct the feature maps set
$Vcb_{set} = \{\underbrace{Vcb_1, Vcb_1, \ldots, Vcb_1}_{K}, Vcb_1, Vcb_2, \ldots, Vcb_{K+1}\}$ ▷ Construct the Vcb set
for $i = 1; i++; i \leq \infty$ **do**
  for $j = max(i, i-K); j++; j \leq i+K$ **do**
    $f_{j \to i} = \mathcal{W}(f_j, \mathcal{F}(I_i, I_j))$ ▷ Warp the feature to the reference frame
  end for
  $\omega_{set} = softmax(L2norm(Vcb_{set}))$ ▷ Calculate ω for each frame
  $\omega_j = \omega_{set}(j)$ ▷ Select the corresponding weight for each frame
  $\bar{f}_i = \sum_{j=i-K}^{i+K} \omega_j f_{j \to i}$ ▷ Feature aggregation for the reference frame
  $y_i = N_{det}(\bar{f}_i)$ ▷ Detection result for the reference frame
  $f_{i+K+1} = N_{feat}(I_{i+K+1})$ ▷ Update feature and Vcb buffer
  $M_{blur\_i+K+1} = N_{blur}(I_{i+K+1})$
  $M_{saliency\_i+K+1} = N_{saliency}(I_{i+K+1})$
  $M_{blur\_cali\_i+K+1} = M_{blur\_i+K+1} \otimes M_{saliency\_i+K+1}$
  $Vcb_{i+K+1} = \Sigma u(M_{blur\_cali\_i+K+1})$
  $Vcb_{set} = \{Vcb_{i+1}, Vcb_{i+2}, \ldots, Vcb_{i+K+1}\}$
end for

IV. EXPERIMENT

A. EXPERIMENT SETUP

We evaluate the proposed method on the prevalent large-scale dataset for video object detection, ImageNet VID dataset. It contains 3862 training sets, 555 validation sets and 937 test sets, and they have been well fully annotated. There are 25 or 30 frames in most video snippets. Following the strategy in [22], we implement our training on the combination of the DET training set and the VID training set. The DET training set contains 200 classes. the VID training set contains 30 classes, which is a subset of the categories in the DET training set. We hence only extract the same 30 classes annotations in the DET trainning set for training. The validation set is used for mean average precision (mAP) evaluation.

We train the proposed model using PyTorch [42] framework on a PC with one Xeon E5-25678 v2 @2.50GHz CPU and four NVIDIA 2080Ti GPU. The input images are all resized to 600 pixels for the shorter sides. The model is traned on 4 GPUs. Each GPU holds only one mini-batch and each mini-batch contains one sets of images for one reference frame. We utilize SGD to optimize the network for totally 120K iterations. The learning rate is set as 1×10^{-3} for the first 80K iterations and 1×10^{-4} for the last 40K iterations. In the training phase, we take random two frames in the $2K + 1$ ranges to increase the robustness. In the inference phase, we set $K = 9$, that is the features of 19 frames are aggregated for the reference frame detection. We abandon the post-processing methods such as Seq-NMS to refine the detection results for simplicity, because it is not our emphasis.

B. RESULTS

We compare the proposed method with the state-of-the-art video object detection algorithms as shown in Table 1. The algorithms listed in Table 1 all use ResNet-101 as the feature extraction backbone. Moreover, no post-processing steps such as Seq-NMS are utilized for fairness. The proposed method is modified based on FGFA [22]. Compared to the baseline FGFA, BFAN makes progress in 25 categories and improves 2.8% for all classess. D&T [6] combine the detection and the tracking algorithms, and it falls behind other end-to-end algorithms except DFF [21]. DFF is designed for high speed, thus it sacrifices the precision. MANet [23] is also based on FGFA, which introduces the instance level feature aggregation. SCNet [43] achieves 77.9% with the scale-aware module and the coupling-structure ROI module. However, BFAN still outperforms MANet and SCNet with only the object blur evaluation.

TABLE 1. Quantitative results on ImageNet VID validation set. The mAP for each class in VID dataset is listed and as well as the mAP for all classes. The feature extraction backbone is denoted as N_{feat}. R101 is short for ResNet-101.

Method	airplane	antelope	bear	bicycle	bird	bus	car	cattle
Faster R-CNN [3]	90.5	80.1	83.0	69.6	73.4	72.4	57.2	62.5
D&T [6]	89.4	80.4	83.8	70.0	71.8	82.6	56.8	71.0
DFF [21]	84.6	82.1	84.1	67.1	71.1	76.1	56.5	67.8
FGFA [22]	89.4	85.1	83.9	69.8	73.5	79.0	60.6	70.7
MANet [23]	90.1	87.3	83.4	70.9	73.0	75.6	62.0	74.0
SCNet [43]	90.3	76.3	84.3	75.7	75.8	84.3	62.9	67.9
BFAN	91.0	80.6	86.2	76.8	75.9	83.4	65.5	73.9
Method	dog	cat	elephant	fox	g_panda	hamster	horse	lion
Faster R-CNN [3]	69.0	81.6	77.3	85.0	80.7	87.0	72.5	41.6
D&T [6]	71.8	76.6	79.3	89.9	83.3	91.9	76.8	57.3
DFF [21]	65.0	82.3	76.3	87.8	81.9	91.3	70.3	47.4
FGFA [22]	72.5	84.3	79.9	89.8	81.0	93.3	72.3	50.5
MANet [23]	73.3	85.3	79.6	91.6	83.5	96.5	74.5	70.5
SCNet [43]	72.4	87.0	82.4	91.5	83.0	93.0	75.5	64.5
BFAN	76.7	87.8	79.6	89.9	87.9	95.9	77.6	67.0
Method	lizard	monkey	motor	rabbit	red_panda	sheep	snake	squirrel
Faster R-CNN [3]	78.0	52.2	81.2	66.6	81.5	57.3	70.5	53.1
D&T [6]	79.0	54.1	80.3	65.3	85.3	56.9	74.1	59.9
DFF [21]	76.5	45.7	78.1	62.8	77.8	55.8	74.5	50.5
FGFA [22]	80.8	52.3	83.0	72.7	84.0	57.8	77.1	55.8
MANet [23]	82.0	54.4	81.6	67.0	89.3	73.3	77.4	54.3
SCNet [43]	83.0	54.4	84.7	73.0	81.5	72.0	81.0	54.6
BFAN	79.8	55.6	86.0	67.8	83.1	63.6	83.7	57.2
Method	tiger	train	turtle	watercraft	whale	zebra	mAP(%)	N_{feat}
Faster R-CNN [3]	90.8	82.3	79.1	64.6	75.0	91.2	73.4	R101
D&T [6]	91.3	84.9	81.9	68.3	68.9	90.9	75.8	R101
DFF [21]	90.2	81.7	77.9	65.8	66.2	89.5	72.8	R101
FGFA [22]	91.9	83.8	83.3	68.7	75.9	91.1	76.5	R101
MANet [23]	91.9	82.9	80.3	69.3	75.4	92.4	78.1	R101
SCNet [43]	91.8	82.1	79.9	68.3	74.5	90.1	77.9	R101
BFAN	92.7	85.9	81.1	69.8	78.6	91.7	**79.1**	R101

C. EXECUTION TIME

We test the execution time on ImageNet VID dataset. We select 100 images with the resolution of 1280 × 720, and calculate the average execution time for processing the 100 images. Different from the training phase, we test the execution time on a PC with an Intel CPU i5-9600K@3.7GHz, 16GB RAM and one NVIDIA 1080 GPU. The execution time only includes the process of running the network without other processes such as decoding input images. As shown in Table 2, Faster R-CNN and DFF only consider the reference frame, thus they are faster. BFAN and FGFA both take 19 frames as the supporting frames for the reference frame. The process of feature aggregation takes more time compared to the still image detector. BFAN is based on FGFA by adding the blur evaluation guided weights calculation. As a result, BFAN improved 2.8% with only 3% more running time, which is valuable.

TABLE 2. Execution time for an image with the resolution of 1280 × 720 on a PC with an Intel CPU i5-9600K@3.7GHz, 16GB RAM and one NVIDIA 1080 GPU. We calucate the average execution time on testing 100 images.

Method	mAP(%)	Time(ms)
Faster R-CNN [3]	73.4	503
DFF [21]	72.8	375
FGFA [22]	76.3	1333
BFAN	79.1	1372

D. ABLATION STUDY

Table 3 lists the comparison among the proposed method and its different variants. Because we modify the network based on FGFA, we list the performance of FGFA as Method (b). Moreover, we replace the adaptive weight with the average accumulation as Method (a) for comparison. Noteworthy, the item "end-to-end training" only refers to the loaded pre-trained parameters of the blur mapping network and the saliency detection network.

Method (a) is a variant of FGFA. We remove the adaptive weight part and assign each feature with $\frac{1}{2K+1}$. We find that mAP for all categories decrease little by only 0.2%, which indicates that the adaptive weight has a limited effect. The adaptive weight improves the detection precision significantly for the fast motion cases, but it brings a little drop for the slow and medium motion cases. Therefore, some existing video object detection algorithms [23] adopt the simple average accumulation instead of adaptive weight for simplicity.

Method (b) is FGFA. Compared to Method (a), FGFA increased the detection precision for all categories by only

TABLE 3. The performance of different variants of BFAN on ImageNet VID validation sets. All variants are based on the ResNet-101 feature extraction backbone and Faster R-CNN detection network. The relative gains compared to the baseline (a) are shown in the subscript.

Net_{feat}: ResNet-101, Net_{det}: Faster R-CNN						
Method	(a)	(b)	(c)	(d)	(e)	(f)
adaptive weights?		✓				
blurMap?			✓		✓	✓
saliencyMap?				✓	✓	✓
end-to-end training?(blur & saliency)	-	-	✓	✓		✓
mAP(%)	76.1	$76.3_{\uparrow 0.2}$	$78.5_{\uparrow 2.4}$	$78.2_{\uparrow 2.1}$	$78.5_{\uparrow 2.4}$	$79.1_{\uparrow 3.0}$
mAP(%)(slow)	84.2	$83.5_{\downarrow 0.7}$	$86.1_{\uparrow 1.9}$	$85.8_{\uparrow 1.6}$	$87.0_{\uparrow 2.8}$	$86.9_{\uparrow 2.7}$
mAP(%)(medium)	76.2	$75.8_{\downarrow 0.4}$	$77.1_{\uparrow 0.9}$	$76.7_{\uparrow 0.5}$	$76.4_{\uparrow 0.2}$	$77.2_{\uparrow 1.0}$
mAP(%)(fast)	52.1	$57.6_{\uparrow 5.5}$	$54.9_{\uparrow 2.8}$	$55.0_{\uparrow 2.9}$	$54.4_{\uparrow 2.3}$	$55.0_{\uparrow 2.9}$

0.2%. It indicates that the most important part of FGFA is flow motion guided feature aggregation, and the adaptive weight is limitedly effective. We hence preserve the flow motion guided feature aggregation backbone, and propose a more effective method for weight assignment.

Method (c) introduces the blur mapping network to calculate the blur degree of each frame. The pre-trained parameters of the blur mapping network is optimized in the training process. Although Method (c) underperforms FGFA for the fast motion, it improved 2.4% for all categories compared to the adaptive weight in FGFA.

Method (d) introduces the saliency detection network to alleviate the background interference. Method (d) is able to focus on the region of interest, i.e. objects with the help of saliency detection network. Method (d) also improved the performance compared Method (a) and Method (b), which indicates that the background interference is harmful to detection.

Method (e) adopts both the blur mapping network and the saliency detection network. However, the pre-trained parameters of these two sub-networks are frozen in the training process. Although the parameters of these sub-networks cannot be optimized for video object detection task, Method (e) still outperforms Method (c) and Method (d) who only use either blur mapping network or saliency detection network. It can be inferred that the combination of blur mapping and saliency detection achieves the goal of object blur degree evaluation without background interference.

Method (f) is the proposed BFAN method, which unfreeze the pre-trained parameters of the blur mapping network and the saliency detection network based on Method (e). It increases the mAP score by 3% to 79.1% compared to Method (a). The improvement for slow motion and fast motion cases are both significant, which indicates that BFAN is more balance than FGFA.

To sum up, aggregating the feature maps from ajacent frames guided by the object blur degree evaluation is more effective than the adaptive weight module in FGFA. The combination of blur mapping network and the saliency detection network achieves the goal of alleviating the background interference. Through above the modules, the mAP for all categories is improved by 2.8% to 79.1%.

TABLE 4. Results of different frames in testing phase.

testing frame	3	7	11	15	19	23
mAP(%)	73.8	75.0	76.0	77.0	**79.1**	79.2
Time(ms)	808	885	1012	1258	1372	1466

E. AGGREGATION FRAMES ANALYSIS

We exploit the influence the number of the supporing frames in the testing phase as shown in Table 4. We tried 3, 7, 11, 15, 19, 23 frames in inference using 2 frames in training and ResNet-101 as backbone. As expected, the detection accuracy improves with the increased aggregated frames in inference. However, the execution time also increases with more frames are taken into consideration. Results in Table 4 show that the improvement saturates at 23 frame with much more time taken. We hence select 19 frames for the balance between accuracy and running speed.

F. QUALITATIVE RESULTS

Figure 4 shows the visual examples on ImageNet VID datasets. We list three methods including Faster R-CNN (still image detector), FGFA (video detector baseline) and the proposed BFAN. In theory, FGFA intorduces the flow motion compensation and adaptive weight module into Faster R-CNN, the proposed BFAN replaces the adaptive weight module with the weight guided by object blur evaluation. Faster R-CNN detects the incorrect "bicycle" in the first two frames, and FGFA also fails in (b) and (c). The proposed BFAN not only detects the correct "motorcycle" in all five frames, but also gives very high confidence scores compared to Faster R-CNN and FGFA.

G. LIMITATION

The proposed method may fail when the object in the input frame is too blurry to be recognized. As shown in Figure 5, the dog in the top row becomes more and more blurry from left to right. Although the proposed BFAN method succeeds in the first frame, it detects a squirrel by mistake in the second frame. The dog in the third frame is too blurry, which is difficult for BFAN to give the correct detection result. The core idea of BFAN is to adopt the strong features of the

FIGURE 4. The visual examples on ImageNet VID validation sets. (a)–(e) are consecutive frames from one video clips. From top to bottom, three rows shows the detection results of Faster R-CNN, FGFA, BFAN, respectively. These three methods all use ResNet-101 as feature extraction backbone. The dark green boxes labeled "motorcycle" are correct, and the light green boxes labeled "bicycle" are incorrect. The proposed BFAN method outperforms other two object detection methods.

FIGURE 5. Limitation of the proposed BFAN method. (a), (b) and (c) are three frames in serial but not consecutive. The top row shows the failure cases due to the too blurry appearance. The bottom row shows the failure cases due to the occlusion. The frames in the first column both show the correct result, and the frames in the latter two columns all show the failure cases.

clear object appearance in adjacent frames to make up the weak features of the current frame. However, BFAN may fail to give the correct results when most object appearance in adjacent frames is too blurry, which cannot support the strong features. Another failure case is due to the severe occlusion as shown in the bottom row in Figure 5. The zebra in the first frame is detected successfully, but the zebras in the latter two frames are occluded by the pillar. The BFAN fails to detect the zebra due to the weak features affected by the occlusion. The possible solver is training the network with more blurry object cases and occluded cases to improve the robustness.

V. CONCLUSION

In this article, we propose a video object detection algorithm guided by the object blur degree evaluation. We improve the weight assignment for the aggregated frames with the blur prior. Especially, a blur mapping network is introduced to label each pixel as either blur or non-blur. Because we only care about the object blur degree without the background, a saliency detection network is adopted to focus on the objects. Calibrated by the saliency map, the calibrated blur map which focus on object blur degree is obtained to calculate the weight for each frame. The extensive experiments demonstrate that the proposed method outperforms state-of-the-art video object detection algorithms with affordable increased computation. However, the blur mapping and saliency networks may fail for some unusual cases that the objects are too small to be distinguished, which can be improved in the future work. Futhermore, another important degenerate element in video object detection is rare poses.

We will design special module to tackle rare poses in the future. It is beneficial to video object detection accuracy improvement.

REFERENCES

[1] X. Wu, D. Hong, J. Chanussot, Y. Xu, R. Tao, and Y. Wang, "Fourier-based rotation-invariant feature boosting: An efficient framework for geospatial object detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 2, pp. 302–306, Feb. 2020.

[2] X. Wu, D. Hong, J. Tian, J. Chanussot, W. Li, and R. Tao, "ORSIm detector: A novel object detection framework in optical remote sensing imagery using spatial-frequency channel features," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 5146–5158, Jul. 2019.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[4] Y. Li, S. Li, C. Chen, A. Hao, and H. Qin, "Accurate and robust video saliency detection via self-paced diffusion," *IEEE Trans. Multimedia*, vol. 22, no. 5, pp. 1153–1167, May 2020.

[5] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 817–825.

[6] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3057–3065.

[7] C. Chen, G. Wang, C. Peng, X. Zhang, and H. Qin, "Improved robust video saliency detection based on long-term spatial-temporal information," *IEEE Trans. Image Process.*, vol. 29, pp. 1090–1100, 2020.

[8] C. Chen, J. Song, C. Peng, G. Wang, and Y. Fang, "A novel video salient object detection method via semi-supervised motion quality perception," 2020, *arXiv:2008.02966*. [Online]. Available: http://arxiv.org/abs/2008.02966

[9] Y. Li, S. Li, C. Chen, A. Hao, and H. Qin, "A plug-and-play scheme to adapt image saliency deep model for video data," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Sep. 10, 2020, doi: 10.1109/TCSVT.2020.3023080.

[10] C. Mu, J. Liu, Y. Liu, and Y. Liu, "Hyperspectral image classification based on active learning and spectral-spatial feature fusion using spatial coordinates," *IEEE Access*, vol. 8, pp. 6768–6781, 2020.

[11] X. Wu, W. Li, D. Hong, J. Tian, R. Tao, and Q. Du, "Vehicle detection of multi-source remote sensing data using active fine-tuning network," *ISPRS J. Photogramm. Remote Sens.*, vol. 167, pp. 39–53, Sep. 2020.

[12] G. Batchuluun, Y. W. Lee, D. T. Nguyen, T. D. Pham, and K. R. Park, "Thermal image reconstruction using deep learning," *IEEE Access*, vol. 8, pp. 126839–126858, 2020.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[15] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[19] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[21] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep feature flow for video recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4141–4150.

[22] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 408–417.

[23] S. Wang, Y. Zhou, J. Yan, and Z. Deng, "Fully motion-aware network for video object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 557–573.

[24] D. Liu, Y. Cui, Y. Chen, J. Zhang, and B. Fan, "Video object detection for autonomous driving: Motion-aid feature calibration," *Neurocomputing*, vol. 409, pp. 1–11, Oct. 2020.

[25] F. Xiao and Y. J. Lee, "Video object detection with an aligned spatial-temporal memory," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 494–510.

[26] Z. Jiang, P. Gao, C. Guo, Q. Zhang, S. Xiang, and C. Pan, "Video object detection with locally-weighted deformable neighbors," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8529–8536.

[27] G. Bertasius, L. Torresani, and J. Shi, "Object detection in video with spatiotemporal sampling networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 342–357.

[28] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Single shot video object detector," *IEEE Trans. Multimedia*, early access, Apr. 23, 2020, doi: 10.1109/TMM.2020.2990070.

[29] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[30] J. Shi, L. Xu, and J. Jia, "Discriminative blur detection features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2965–2972.

[31] X. Yi and M. Eramian, "LBP-based segmentation of defocus blur," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1626–1638, Apr. 2016.

[32] Y. Pang, H. Zhu, X. Li, and J. Pan, "Motion blur detection with an indicator function for surveillance machines," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5592–5601, Sep. 2016.

[33] K. Ma, H. Fu, T. Liu, Z. Wang, and D. Tao, "Deep blur mapping: Exploiting high-level semantics by deep neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5155–5166, Oct. 2018.

[34] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[36] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1647–1655.

[37] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.

[38] C. Chen, S. Li, H. Qin, Z. Pan, and G. Yang, "Bilevel feature learning for video saliency detection," *IEEE Trans. Multimedia*, vol. 20, no. 12, pp. 3324–3336, Dec. 2018.

[39] C. Chen, S. Li, Y. Wang, H. Qin, and A. Hao, "Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3156–3170, Jul. 2017.

[40] G. Ma, S. Li, C. Chen, A. Hao, and H. Qin, "Stage-wise salient object detection in 360° omnidirectional image via object-level semantical saliency ranking," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 12, pp. 3535–3545, Dec. 2020.

[41] S.-H. Gao, Y.-Q. Tan, M.-M. Cheng, C. Lu, Y. Chen, and S. Yan, "Highly efficient salient object detection with 100k parameters," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 702–721.

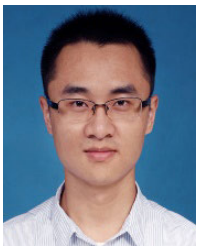
[42] P. Adam, G. Sam, C. Soumith, C. Gregory, Y. Edward, D. Zachary, L. Zeming, D. Alban, A. Luca, and L. Adam, "Automatic differentiation in pytorch," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[43] F. Wang, Z. Xu, Y. Gan, C.-M. Vong, and Q. Liu, "SCNet: Scale-aware coupling-structure network for efficient video object detection," *Neurocomputing*, vol. 404, pp. 283–293, Sep. 2020.

YUJIE WU received the B.S. degree from Beihang University, Beijing, China, in 2014, where he is currently pursuing the Ph.D. degree with the Image Processing Center. His research interests include image deblurring, object detection, and computer vision.

HONG ZHANG received the B.S. degree from the Hebei University of Technology, China, the M.S. degree from the Harbin University of Science and Technology, China, and the Ph.D. degree from the Beijing Institute of Technology, China, in 1988, 1993, and 2002, respectively, all in electrical engineering. She was a Visiting Scholar with the Department of Neurosurgery, University of Pittsburgh, from 2007 to 2008. She is currently a Professor with the School of Astronautics, Beihang University, Beijing, China. Her research interests include activity recognition, image restoration, image indexing, object detection, and stereovision.

YAWEI LI received the B.S. degree from Xidian University, Xi'an, China, in 2013. He is currently pursuing the Ph.D. degree with the Image Processing Center, Beihang University, Beijing, China. His research interests include computer vision and machine learning, with an emphasis on image restoration and image deblurring.



YIFAN YANG received the M.S. degree from the School of Automation and Electrical Engineering, University of Science and Technology Beijing, China, in 2011, and the Ph.D. degree from the Image Processing Center, School of Astronautics, Beihang University, Beijing, China, in 2018. He is currently a Postdoctoral Researcher with Beihang University. His research interests include image denoising, image enhancement, image registration, real-time image processing, pattern recognition, and embedded systems.

DING YUAN (Member, IEEE) received the B.Eng. degree in measurement, control technology and instrument from the Harbin Institute of Technology, China, in 2001, and the M.Phil. and Ph.D. degrees in mechanical and automation engineering from The Chinese University of Hong Kong in 2004 and 2008, respectively. She joined the School of Astronautics, Beihang University, in 2009, where she is currently an Assistant Professor with the Image Processing Center. Her research interests include stereo vision, 3-D reconstruction, camera calibration, and camera's ego-motion estimation.