IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 27, 2018, accepted April 27, 2018, date of publication May 24, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2836921

MPNET: An End-to-End Deep Neural Network for Object Detection in Surveillance Video

HANYU WANG[1], PING WANG[1], AND XUEMING QIAN[1,2], (Member, IEEE)
[1]School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, 710049, China
[2]ZhiBiAn Technology, Taizhou 317000, China

Corresponding authors: Ping Wang (ping.fu@mail.xjtu.edu.cn) and Xueming Qian (qianxm@mail.xjtu.edu.cn)

This work was supported in part by NSFC under Grant 61732008, Grant 61772407, and Grant u1531141, in part by the National Key Research and Development Program of China under Grant 2017YFF0107700, and in part by the Guangdong Provincial Science and Technology Plan Project under Grant 2017A010101006.

ABSTRACT Object detection is one of the most important topics in computer vision task and has obtained impressive performance thanks to the use of deep convolutional neural network. For object detection, especially in still image, it has achieved excellent performance during past two years, such as the series of R-CNN which plays a vital role in improving performance. However, with the number of surveillance videos increasing, the current methods may not meet the growing demand. In this paper, we propose a new framework named moving-object proposals generation and prediction framework (MPGP) to reduce the searching space and generate some accurate proposals which can reduce computational cost. In addition, we explore the relation of moving regions in feature map of different layers and predict candidates according to the results of previous frames. Last but not least, we utilize spatial-temporal information to strengthen the detection score and further adjust the location of the bounding boxes. Our MPGP framework can be applied to different region-based networks. Experiments on CUHK data set, XJTU data set, and AVSS data set, show that our approach outperforms the state-of-the-art approaches.

INDEX TERMS Object detection, motion-probed proposals, proposals prediction, surveillance video, deep neural network.

I. INTRODUCTION

Object detection, which is a classic branch of computer vision tasks, plays an increasingly important role in the system of intelligent video surveillance, automotive safety and robotics. With the increasing number of surveillance videos, fast and accurate object detection is urgently needed. Although many researchers have proposed some solutions in the past decade, there are still some challenges in object detection, such as motion blur, video defocus and etc.

Recently, the performance of object detection is significantly improved thanks to the use of deep convolutional neural network (CNN) [12]–[14], [23], especially, with the development of the series of R-CNN [1]–[4] approaches. These approaches integrate proposals generation into the network which speed up the detection and achieve end-to-end training. For example, the region proposal network (RPN) in Faster R-CNN replaces the Selective Search [9] and shares convolutional layers with detection networks. RPN starts with exhaustive search essentially and it is not effective to detect low-resolution objects. Moreover, these methods are designed for object detection in still image. With the increasing number of surveillance videos, how to robustly and effectively detect objects in surveillance videos becomes a problem, which demands a prompt solution.

Existing methods are committed to simplify the network to speed up the detection [7], [8], [18] or to redesign networks for extracting more robust features [4]. These performances depend on the accuracy of proposals generation to a large ext ent. For example, in [7], Redmon *et al.* treat object detection as a regression problem and they predict proposals in each grid. Although this method can speed up the detection process, the accuracy of bounding boxes is lower than the RPN [3], [4] based approaches. Most importantly, these methods cannot detect low-resolution objects. Furthermore, since object detection is the basis of object re-identification, the accuracy of the bounding boxes is significantly important. In addition, the detection scores of the same pedestrian in adjacent frames may fluctuate which will affect the object detection results. Motion blur, part occlusion, background change or video defocus may cause the fluctuation of detection score. Hence, how to get stable object detection performance in video sequences is a great challenge.

Recently, the task on the object detection from video is proposed. In [6], a framework for VID (ImageNet task on object detection from video) was proposed. It combines still image object detection with tracking for tubelets. However, this framework is computationally expensive and time-consuming. Because they not only detect objects in still image which is implemented by R-CNN, but also track the objects with high scores, which we call high-confidence objects.

This paper is motivated by the fact that current methods [2], [3], [7], [8] often miss low-resolution objects and the process of generating proposals are based on exhaustive search which is resource consuming. We can use the spatial-temporal information to generate suitable proposals for object detection. Inspired by Faster R-CNN, we find that the process of mapping proposals to feature map can save a significant amount of time. Nevertheless, the RPN generates proposals based on the full image at each frame, which is a heavy workload. Moreover, Huang *et al.* [27] verified that reducing the number of region proposals will speed up the detection process and will not harm the detection a lot.

Considering that most of surveillance videos have fixed background, therefore, we propose a Moving-Object Proposals Generation and Prediction Framework (MPGP). We design MPFP framework to generate accurate motion-probed proposals and predict the locations of objects where they are likely to occur in the next frame. Furthermore, we adjust the locations and the scores of bounding boxes in conjunction with the motion information and the spatial-temporal information between adjacent frames.

The main contributions are summarized as follows:

1) We propose a network named MPNET for object detection in surveillance video, which is a deep neural network for end-to-end object detection. Various objects such as vehicles and pedestrians can be detected in a unified framework.

2) We propose a moving-object proposals generation and confidence-based proposal prediction framework to find precise proposals for object detection. The spatial-temporal information in different feature maps is explored to obtain the proposals of moving objects. It has the advantage to probe small moving objects. The confidence-based proposal prediction can reinforce object detection performance by fusing the detection results in adjacent frames, which is robust to the detection result variations. In our proposal generation framework, an adjustment process is proposed to accurately localize objects. This makes our framework robust to the objects with problems of low resolution and confidence fluctuation etc.

3) Our framework can be applied to different kinds of region-based deep neural networks, such as Faster R-CNN and PVANET etc. Our experiments show that our MPNET achieves state-of-art performance compared to other methods. In addition, we propose an algorithm of refining the initial detection results. In our proposed algorithm, we utilize the relation between motion-probed bounding boxes and predicted bounding boxes to suppress the false positives and adjust the results of the true positives more accurately.

II. RELATED WORK

In the past decades, the hand-crafted features [15], [16], [25], [26] have achieved great performance. For example, the classic DPM algorithm [15] proposed a multi-scale deformable part model, which achieved excellent performance in 2009. However, with the development of CNN, deep features, which learn from raw pixels, have demonstrated superior performance. State-of-art object detection methods are always based on deep convolutional neural network [12]–[14], [23]. Current object detection methods can be divided into two parts: region-based methods [1]–[5] and region-free methods [7], [8], [18].

The series of R-CNN [1]–[4] are region-based detection methods which provided better solutions to robust object detection. These methods can be classified into three parts: CNN feature extraction, region proposals generation and classification. Girshick *et al.* [1] proposed R-CNN for training deep convolutional feature to classify region-based proposals. Fast R-CNN [2] fixes the disadvantage of R-CNN [1] and SPPnet [5], which combined feature extraction, classification and regression in a network for training. However, Fast R-CNN still uses Selective Search [9] to generate proposals. Although some optimized methods for proposals generation were proposed [10], the network is still not end-to-end. Therefore, Faster R-CNN [3] is proposed to solve this problem. They proposed a Region Proposals Network (RPN) to replace the Selective Search, which achieved end-to-end training and test. Moreover, not only the performance but also the speed of object detection also improved a lot. Recently, some variants [4], [11] have been proposed to further improve the performance of Faster R-CNN, which are based on Faster R-CNN. For example, in [11], multi-scale feature maps are combined for object detection. Hong *et al.* [4] redesigned the feature extraction part in Faster R-CNN with the principle of ‘‘less channels with more layers’’. Moreover, concatenated rectified linear unit (C.ReLU) [19] and inception [20] are used to reduce the amount of computation.

Despite the better performance of object detection has achieved, the detection process is also computational intensive. So, some region-free methods, such as YOLO [7], [18] and SSD [8], are proposed to speed up the detection process. YOLO [7] divides an image into regular grids and predicts locations based on these grids. Although the speed of YOLO is faster than [1]–[4], its performance is worse than the performance of Faster R-CNN. What's worse, the methods based on grid prediction will miss the objects if multiple objects are in a grid. To solve this problem, SSD [8] combines different size feature maps that generated by different convolutional layers. In addition, it used small convolutional filters to predict category score and box offsets. Kong *et al.* [24] proposed a Reverse Connection with Objectness Prior Networks to combine the region-based (e.g., Faster R-CNN) and region-free methods (e.g., SSD). In [22], a deep feature pyramid network is proposed to solve multi-scale problems.

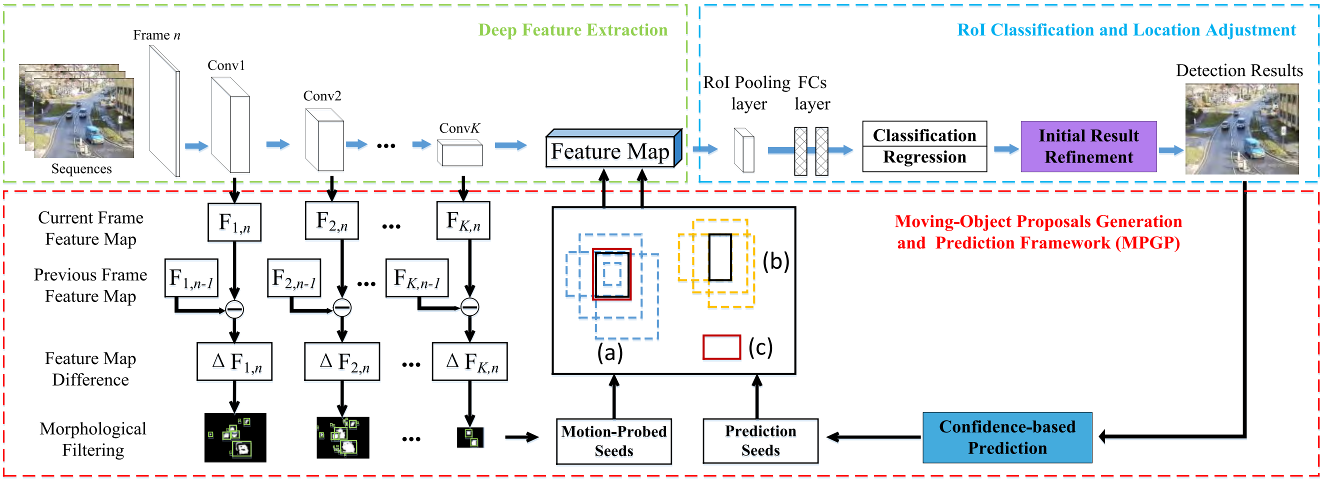

FIGURE 1. Overview of MPGP-based Object Detection Network (MPNET). We propose MPGP framework to generate more accurate proposals and use the spatial-temporal information to refine the initial detection results. We use black boxes to denote the motion-probed seeds and use red boxes to denote the predicted proposals. The dotted lines denote the adjusted proposals of motion-probed seeds and low-confidence predicted proposals. (a) is an object which generates proposals by moving-object proposals generation and confidence-based proposals prediction at the same time. (b) is an object which generates proposals only by moving-object proposals generation. (c) is an object which generates proposals only by confidence-based proposals prediction.

The methods mentioned above are based on still-image. Recently, the ImageNet challenge on object detection from video brings up a new question on how to solve the object detection problems for videos robustly and effectively. Some researches [6], [30]–[34], [36]–[42] incorporate spatial-temporal information to enhance the performance of object detection in video domain. For example, Kang *et al.* [6], [30] proposed T-CNN to combine still-image object detection with object tracking, which utilizes optical flows to predict bounding boxes to adjacent frame and then get tubelets by tracking high-confidence boxes. This network is multi-stage pipeline. FGFA [36] focuses on improving feature quality through flow-guided feature aggregation. The feature maps from nearby frames are warped to the reference frame according to the flow. However, this method is computational intensive compared to [3], [4], [8], and [18]. Cascaded regional spatio-temporal feature-routing networks (CRFN) [37] is proposed to incorporate the correlation filter tracking on the convolutional feature maps. The context information is utilized via a Look-Up-Table method to suppress the conflicting false positives and guide the detector to produce a semantically coherent interpretation on the video. Huang and Chen [38] and Huang and Do [41] proposed to combine a probabilistic background generation (PBG) module and a moving object detection (MOD) module for moving object detection. The PBG module is proposed to produce the probabilistic background model in variable bit-rate video streams. The MOD module used a block selection procedure to find the blocks belonging to moving objects. In [39] and [42], Chen and Huang proposed an approach to detect moving object in different bit-rate video streams. Temporal dynamic graph Long Short-Term Memory network (TD-Graph LSTM) [40] uses action descriptions as supervision instead of using bounding boxes to obtain the objects of interest. It recurrently propagates the temporal context on a constructed dynamic graph structure for each frame, which helps alleviate the missing label problem. The methods mentioned above are not dedicated to solve the problem of missing small object and motion blur.

Most state-of-art still-image object detection methods are following the common pipeline of "CNN feature extraction + region proposal generation + RoI classification." Faster R-CNN [3] and PVANET [4] achieve state-of-art performance in still-image object detection. However, RPN, which is used to generate proposals in Faster R-CNN and PVANET, always searches proposals based on the full feature map. They are resource consuming and neglect the spatial-temporal information in surveillance videos. Therefore, we mainly redesign the region proposal part.

III. THE PROPOSED APPROACH

A. DEEP OBJECT DETECTION FRAMEWORK OVERVIEW

Our MPGP-based Object Detection Network (MPNET) is illustrated in Fig. 1. Our framework is a region-based method and consists of the following three parts: 1) Deep feature extraction. We fuse feature maps of the deep convolutional layers to detect objects with various resolutions. 2) MPGP-based region proposal generation. We use our MPGP to replace the RPN in other region-based methods to generate some accurate proposals. 3) Region of interest (RoI) classification and location adjustment. We classify the proposals generated by MPGP and adjust location of the bounding boxes to obtain the precise results. More details can be discussed as follows.

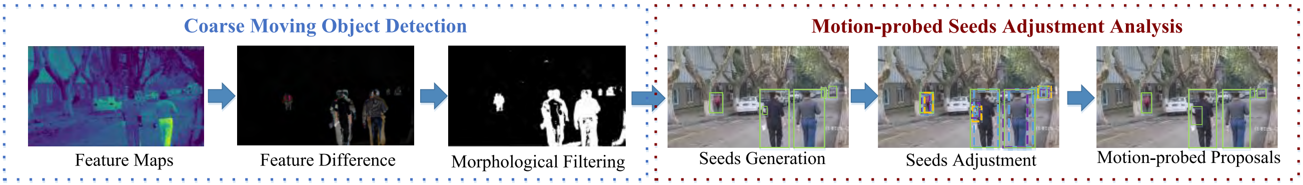

FIGURE 2. The process of generating motion-probed proposals.

B. DEEP FEATURE EXTRACTION

The input of our framework are the sequences of surveillance videos. We extract deep feature using VGG16 or PVANET feature extraction part, respectively. In Faster R-CNN, it utilizes the transformed VGG16 network with 13 convolutional layers and 5 max-pooling layers to extract feature. PVANET mainly redesigns the feature extraction part with the principle of less channels with more layers. Just as Fig.1 shown, our feature extraction is on the convolutional layers (CL). Assume that the total number of CL is K, $K = 13$ for VGG and 16 for PVANET.

We denote the feature map of frame n which is extracted by k-th convolutional layer as $F^i_{k,n}$, i is the channel index of the k-th convolution layer, $k \in [1, K]$. Due to the pooling process in CNN, the feature maps become smaller (with low resolution) with the increasing of depth of the network. The lower layers often contain local features while the deeper layers contain global features. So, we can fuse the feature maps with various resolution in adjacent frames to get various spatial-temporal information for object detection in video.

C. PROPOSALS GENERATION FRAMEWORK

The small objects are more sensitive in lower layers while the large objects can be complete detect in deeper layers. Therefore, we explore multi-scale spatial-temporal information from different convolutional layers.

We propose a MPGP framework to generate some accurate proposals, which replace the RPN in Faster R-CNN and PVANET etc region based CNN. In RPN, The input of MPGP is the feature map produced by the last shared conv layer. Unlike RPN which uses a sliding window sliding in the feature map per pixel [3], [4], our framework generates some accurate proposals according to the motion-probed and predicted information. The exploration of spatial-temporal information is divided into two aspects: moving-object proposals generation and confidence-based proposals prediction.

1) MOVING-OBJECT PROPOSALS GENERATION

In surveillance videos, objects can be divided into stationary objects and moving objects. The proportion of moving objects is relatively large. Therefore, it is essential to utilize motion information to generate motion-probed proposals. This method is more targeted compared to the methods of searching proposals based on the whole image. The process of generating motion-probed proposals is illustrated in Fig. 2. It consists of two parts: coarse moving object detection and motion-probed seeds (regions) adjustment analysis.

a: COARSE MOVING OBJECT DETECTION

For the frame n, we have obtained the feature maps $F^i_{k,n}$ through the forward propagation of a CNN. Then, we obtain coarse seeds of moving objects according to the following steps:

I) We get multi-scale spatial-temporal information based on the corresponding feature map in adjacent two frames as follows.

$$\Delta F^i_{k,n} = F^i_{k,n} - F^i_{k,n-1} \tag{1}$$

where n, k, and i respective denote the frame, the k-th convolutional layer, and the channel index, $k \in [1, K]$.

II) We get the normalized feature difference of the frame n at the k-th layer by averaging all the channels as follows.

$$\Delta F_{k,n} = (\sum_{i=1}^{m} |\Delta F^i_{k,n}|)/m \tag{2}$$

where m denotes the total number of the channels at layer k.

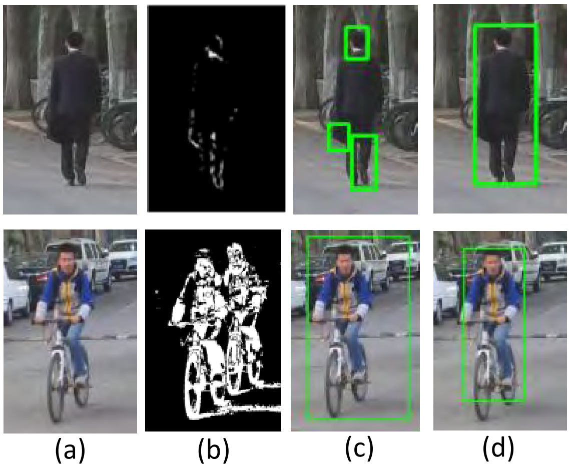

FIGURE 3. The necessity of extending motion seeds. (a) key frame; (b) the binary image of frame difference; (c) motion-probed seeds; (d) adjusted proposals.

III) We further take advantage of morphological filtering to reduce the noise and stress moving areas on $\Delta F_{k,n}$. The steps of morphological filtering are given as follows: <i> We use OTSU to distinguish the foreground of $\Delta F_{k,n}$ from background, as shown in Fig. 3(b). OTSU is a threshold selection method from gray-level histograms. <ii> We carry out media filtering on the binary image to remove isolated noise points. <iii> We dilate the moving areas to stress the edge of the moving objects. <iv> We mark these moving areas with rectangle boxes and map these different size moving areas to raw image as shown in Fig.3(c). We denote these candidates got from coarse motion object detection as motion-seeds. Motion-probed seed is a potential region which has high probability to contain part of an object (including pedestrians, vehicles, etc).

For all the K layers, we carry out the above three steps iteratively. Then we can get the objects at different resolutions.

b: MOTION-PROBED SEEDS ADJUSTMENT ANALYSIS

In surveillance videos, the appearance and the speed of moving objects are diverse. For example, as shown in first row of Fig. 3, the motion-probed seeds are smaller than the pedestrian in black suit. This situation results from color-invariance and small-motion. In this case, we need to enlarge the area of motion-probed seeds and shift the center of the seeds. In another case, as shown in the second row of Fig. 3, the speed of the child riding a bike is fast. Therefore, the motion-probed seed is bigger than the child and the center of the seed shift from the center of the child. Since a motion-probed seed can be a part of an object or an enlarged object, we need to adjust proposals to get more accurate detection results based on the coarse motion-probed seeds.

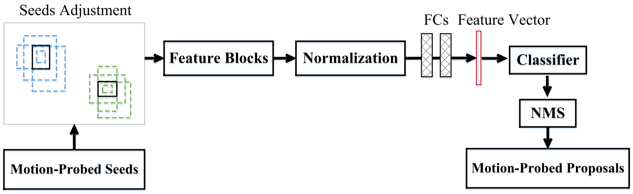

FIGURE 4. The adjustment analysis of motion-probed seeds.

We propose a seeds adjustment method. As shown in Fig. 4, this algorithm is consist of five steps:

I) We change the scale, aspect ratio and the center of motion-probed seeds which are represented by the black solid boxes as shown in the top-left of Fig.4. We use 3 aspect ratios (1, 0.41, 2) [3], [35], 3 scales and 3 different center of seeds. Therefore, we get 27 different transformed proposals.

II) We map those proposals from frame n to the last convolutional feature map (K-th) proportionally to get the feature blocks. More details can be learned in [2].

III) We use max pooling to normalize these feature blocks to a fixed size (e.g., 7∗7). Each feature map channel apply pooling independently.

IV) We use full connected layers to convert those feature blocks to feature vectors to get global features with 4096 dimension.

V) Those feature vectors are fed into a classification layer. The classification layer outputs the probability of the proposals whether they are a foreground or a background.

VI) We use non-maximum suppression (NMS) to filter out redundancy proposals. Since we use multi-scale feature map to obtain motion-probed seeds, some of the proposals represent a same object. Therefore, we need use NMS to merge overlapping proposals for achieving the most accurate proposals. The output is the adjusted motion-probed proposals.

2) CONFIDENCE-BASED PROPOSALS PREDICTION

Due to the motion, illuminance variation, occlusion, pose changing, etc problems, the detection scores (confidences) of the same object in video sequences vary dramatically. As shown in Fig. 5, the confidences of a same object fluctuate dramatically at different frames. The Fig. 5 (a) shows the confidences of a low-resolution object at each frame. We find that most of the confidences are in the range [0.1, 0.5] and only a small faction is higher than 0.5, which are unstable. Moreover, Fig. 5 (b) shows the confidences of a high-resolution object at each frame. Most of confidences are stable between 0.9 and 1 while about 2% scores are smaller than 0.9. From Fig. 5, we find that stable object detection in surveillance videos is a challenge for small objects and objects with occlusion, blur and etc. To solve this problem, we propose a confidence-based proposal prediction method based on the detection results of previous frames. This approach can suppress the influence of confidence fluctuation.

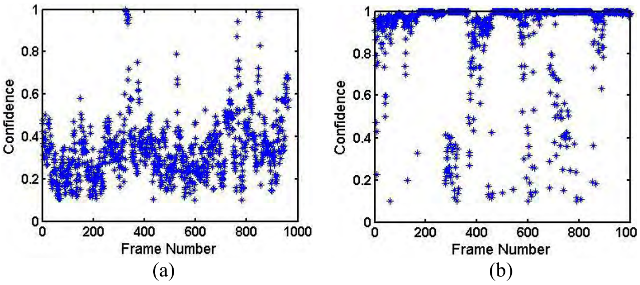

FIGURE 5. The fluctuations of object confidences between adjacent frames.

We propose to utilize the detection results of the two adjacent frames n-1 and n to predict the locations of objects in frame $n + 1$. The diagram is shown in Fig. 6. Based on the proposals' bounding boxes of frames n-1 and n, we get their predicted coordinates of frame $n + 1$. When $n = 1$ and 2, we generate proposals by RPN. When n is greater than 2, if the confidence of the bounding box is high, we predict the location of the bounding box in frame $n + 1$ directly. In another case, if the confidence of the bounding box is low, we need to adjust the box to prevent that the box is only a part of the global or shifts from the object. The process of prediction can be divided into two parts: high-confidence bounding boxes prediction and low-confidence bounding boxes prediction.

a: HIGH-CONFIDENCE BOUNDING BOX PREDICTION

We suppose that the detection result is credible if the confidence is greater than t_s, such as the person in Fig. 6 (a) and the car in Fig.6 (b). Therefore, we only predict the locations of the objects where they would appear in the frame $n+1$ according to their previous frames $n-1$ and n. The predicted width and height of an object can be regarded as a linear transformation from its adjacent frames $n-1$ and n.

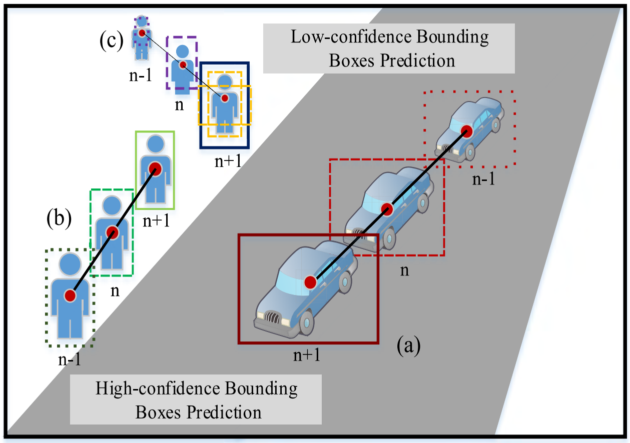
(c)
n-1
n
n+1
Low-confidence Bounding Boxes Prediction
n-1
(b)
n+1
n
n-1
n
n+1
(a)
High-confidence Bounding Boxes Prediction

FIGURE 6. Confidence-based proposals prediction. The distance to the camera and the accuracy of proposals will affect the confidences of objects. We adopt different prediction methods according to the confidence of the objects.

Let $P_{n,b} = \{w_{n,b}, h_{n,b}, cx_{n,b}, cy_{n,b}$ denote the width, height and the coordinates of the center of the b-th bounding boxes. We predict the object location at frame $n+1$ as follows.

$$P_{(n+1),b} = w_{n,b} + \Delta w_{n,b}, h_{n,b} + \Delta h_{n,b}, cx_{n,b} + \Delta cx_b, cy_{n,b} + \Delta cy_b \quad (3)$$

where $\Delta w_{n,b} = w_{n,b} - w_{(n-1),b}$, $\Delta h_{n,b} = h_{n,b} - h_{(n-1),b}$. In addition, $\Delta cx_{n,b} = cx_{n,b} - cx_{(n-1),b}$ and $\Delta cy_b = cy_{n,b} - cy_{(n-1),b}$ are the relative motion of the proposal's center in x and y directions, respectively.

b: LOW-CONFIDENCE BOUNDING BOX PREDICTION

To get robust detection results, especially for low-resolution, occlusion objects, we emphasis the low-confidence objects verification with their confidences are in the range $[t_{min}, t_s]$, such as the person in Fig. 6 (c).

First, we predict the locations of the objects in the next frame according to Eq. (3). Then, we adjust the predicted boxes through changing the scale, aspect ratio and the center of predicted boxes. The adjustment method is same as which we discuss in the section of the motion-probed seeds adjustment analysis.

D. THE RoI CLASSIFICATION AND LOCATION ADJUSTMENT

From the moving-object proposals generation and confidence-based proposals prediction process, we generate a set of object proposals with different sizes. Let $\Re = \{\boldsymbol{R}_n^M, \boldsymbol{R}_n^P\}$ denote the initial detection results of frame n, where R_n^M and R_n^P denote the initial detection results of motion-probed proposals and predicted proposals, respectively. We combine and adjust the initial object detection results to get final results.

We first map those proposals to the last convolutional feature map (K-th) proportionally to get the feature blocks. Then we utilize the RoI pooling layer to convert the feature blocks into a fix size (e.g., $7 * 7$ in [2]). This step is shown in the top-right part of Fig.1. Then, each fixed-size feature map is pushed into two full-connected layers to extract global feature vector with 4096 dimension [3]. Next, this vector is fed into two sibling layer—an object classification layer and a box-regression layer. The box-regression layer outputs 4 dimension coordinate of boxes while the classification layer outputs the probability of objects.

We define $R_{n,i}^M = \left\{l_{n,i}^M, t_{n,i}^M, r_{n,i}^M, b_{n,i}^M\right\}$ and $R_{n,j}^P = \left\{l_{n,j}^P, t_{n,j}^P, r_{n,j}^P, b_{n,j}^P\right\}$, where $\{l_n, t_n, r_n, b_n, s_n\}$ represents the top-left corner (l_n, t_n) and bottom-right corner (r_n, b_n). Let $s_{n,i}^M$ and $s_{n,j}^P$ represent the score of the motion-probed box i and predicted box j, respectively. Let $O_{n,i,j}$ denote the overlap ratio of motion-probed boxes and predicted boxes.

To determine whether the bounding box containing object or not, the following three case is taken into account.

1) We consider that the object in the bounding box is a true positive if the motion-probed box has high overlap rate with the predicted box and at least one score of the two bounding boxes is high. Therefore, we refine the location of the bounding box.

2) We consider the object is a false positive if the overlap ratio is high but the scores are low. In this case, we suppress the detection score.

3) The remaining bounding boxes with high confidence are the complementary part of motion-probed boxes and predicted boxes. We consider they are true positives, such as object (b) and object (c) in Fig. 1.

According to the situations mentioned above, we propose a location adjustment method to improve the performance of the initial detection results. The details are given in Algorithm 1. The specific procedures are as follows:

Step 1: We compute the overlap ratios of motion-probed boxes and predicted boxes according to Eq. 4.

$$O_{n,i,j} = \frac{area(R_n^M \cap R_n^P)}{area(R_n^M \cup R_n^P)} \quad (4)$$

where $area$(x) denotes the area of region x.

If overlap ratios are greater than $\theta(\theta = 0.5)$ and at least one score of the two bounding boxes is greater than t_s, then we go to **Step 2**. If overlap ratios are greater than θ but the scores are both smaller than t_{min}, then we go to **Step 3**. If a bounding box has low overlap ratios with other bounding boxes and its score is greater than t_s, we maintain the coordinates and the scores of the bounding boxes.

Algorithm 1 Initial Results Refinement

```
Input: The initial detection results of motion-probed
       proposals and predicted proposals
Output: The adjusted detection results
N is the number of motion-probed boxes
K is the number of predicted boxes
for $j = 1$ to N do
  for $i = 1$ to K do
    compute the overlap ratio $O_{n,i,j}$
    if $O_{n,i,j} > \theta$ & $(s^P_{n,j} > t_s$ $or$ $s^M_{n,j} > t_s)$ then
      strengthen score and precise location by Eq.5-9
    elif $O_{n,i,j} > \theta \& (s^P_{n,j} < t_{min} \& s^M_{n,j} < t_{min})$ then
      suppress the detection score by Eq.10
    end if
  end for
end for
add the complementary part
```

Step 2: We adjust the location of the object according to the confidences of both $R^M_{n,i}$ and $R^P_{n,j}$ by a weighted linear estimation as follows:

$$l_n = \begin{cases} \alpha * l^M_{n,i} + (1-\alpha) * l^P_{n,j}, & \text{If} s^P_{n,j} > s^M_{n,i} \\ (1-\alpha) * l^M_{n,i} + \alpha * l^P_{n,j}, & \text{If} s^P_{n,j} < s^M_{n,i} \end{cases} \tag{5}$$

$$t_n = \begin{cases} \alpha * t^M_{n,i} + (1-\alpha) * t^P_{n,j}, & \text{If} s^P_{n,j} > s^M_{n,i} \\ (1-\alpha) * t^M_{n,i} + \alpha * t^P_{n,j}, & \text{If} s^P_{n,j} < s^M_{n,i} \end{cases} \tag{6}$$

$$r_n = \begin{cases} \alpha * r^M_{n,i} + (1-\alpha) * r^P_{n,j}, & \text{If} s^P_{n,j} > s^M_{n,i} \\ (1-\alpha) * r^M_{n,i} + \alpha * r^P_{n,j}, & \text{If} s^P_{n,j} < s^M_{n,i} \end{cases} \tag{7}$$

$$b_n = \begin{cases} \alpha * b^M_{n,i} + (1-\alpha) * b^P_{n,j}, & \text{If} s^P_{n,j} > s^M_{n,i} \\ (1-\alpha) * b^M_{n,i} + \alpha * b^P_{n,j}, & \text{If} s^P_{n,j} < s^M_{n,i} \end{cases} \tag{8}$$

$$s_n = \max\{s^P_{n,j}, s^M_{n,i}\} \tag{9}$$

where $\alpha = s_n/(s^P_{n,j} + s^M_{n,i})$.

Step 3: We suppress the initial detection score according to Eq. (10).

$$s_n = min\{s^P_{n,j}, s^M_{n,i}\} \tag{10}$$

IV. EXPERIMENTS

In this section, we evaluate our framework for object detection in surveillance video on AVSS dataset [28], CUHK dataset [21] and our XJTU dataset. In surveillance videos, vehicle and pedestrian are the mainly concerned objects. We evaluate these two classes at different datasets. We evaluate our method for pedestrian detection on two dataset: XJTU Dataset and CUHK Square Dataset with four competitive models: Faster R-CNN [3], YOLO9000 [18], SSD [8], PVANET [4] while vehicle detection on AVSS dataset with five competitive models: Faster R-CNN [3], YOLO9000 [18], SSD [8], PVANET [4], FGFA [36]. These four methods obtain excellent performance on object detection and codes are available. MPNET (Faster R-CNN) and MPNET (PVANET) are our proposed object detection approaches based on Faster R-CNN (VGG16) and PVANET.

Our framework is implemented based on the deep learning framework Caffe and run on a workstation configured with an NVIDIA GTX 1070. We train our network on PASCAL VOC2007.

A. DATASETS

1) CUHK SQUARE DATASET [21]

It is a traffic video sequence of which lasts 60 minutes long and is recorded by a stationary camera. The resolution of frames is 720*576. Wang *et al.* [21] provide some ground truth of pedestrians at some sampled frames, which is consist of 352 images for train and 100 images for test. However, there are some error labels in public ground truth. In addition, some pedestrians are missing in public ground truth. Therefore, we modify the ground-truth and add more annotations of more frames. We mark 3622 frames for pedestrian detection task.

2) AVSS DATASET [28]

It is a surveillance video sequence of traffic road. The resolution of frames is 720*576 pixels. The video sampling rate is 25Hz. We mark 1008 frames for vehicle detection task.

3) XJTU DATASET

This dataset is collected by Smiles LAB. We collected surveillance videos of six representative places in campus for pedestrian detection and retrieval task. Every scene has five sequences and each video sequence lasts 10 minutes. The resolution of the video is 1080p, 20fps. We mark 36,000 frames for pedestrian detection task. This dataset contains the problem of motion blur, occlusion and low-resolution, which is a new challenge for object detection in surveillance video.

B. EVALUATION OF OBJECT DETECTION

The object detection evaluation criterion is same as PASCAL VOC object detection [17]. Our task is judged by precision/recall curve. The principal quantitative measure is the average precision (AP). The AP summaries the shape of the precision-recall curve and calculate as Eq. 11-12. The overlap is set as 0.5 to evaluate the true positives.

$$AP = \frac{1}{11} \sum\nolimits_{r \in \{0, 0.1, \ldots, 1\}} p_{interp}(r) \tag{11}$$

$$p_{interp}(r) = \max_{\tilde{r}: \tilde{r} \geq r} p(\tilde{r}) \tag{12}$$

where $p(\tilde{r})$ is the measured precision at recall $\tilde{r}$.

C. COMPARISON RESULTS

1) PERFORMANCE ON PEDESTRIAN DETECTION

The proposed framework is evaluated on the CUHK Square dataset and XJTU dataset. Table 1 and Fig. 7 show the overall experimental results on CUHK Square dataset. From Table 1, we can see that our framework outperforms other

TABLE 1. Detection results on CUHK dataset.

Methods	AP	Time
Yolo9000	26.85%	0.03s
SSD300	53.79%	0.03s
SSD512	59.79%	0.07s
PVANET	46.80%	0.167s
MPNET(PVANET)	65.13%	0.152s
Faster R-CNN	63.10%	0.19s
MPNET(Faster R-CNN)	67.42%	0.18s

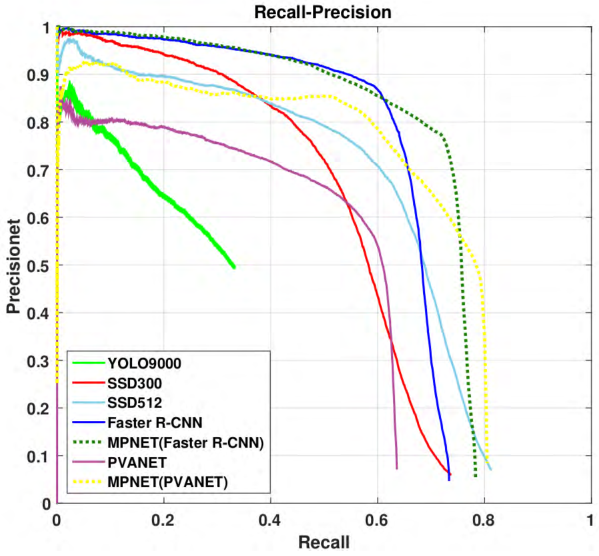

FIGURE 7. The recall-precision curve of pedestrian detection in CUHK dataset.

algorithms. In aspect of average precision, our method gains of 4.32% compared to the Faster R-CNN and gains of 18.33% compared to PVANET. In addition, our MPNET method based on Faster R-CNN obtains the state-of-art performance compared to SSD and YOLO9000. In aspect of computing time, our method achieves a slight decrease compared to Faster R-CNN and PVANET. We achieve a balance between average precision and detection speed.

TABLE 2. Detection results on XJTU dataset.

Methods	AP	Time
Yolo9000	33.09%	0.015s
SSD300	52.02%	0.05s
SSD512	58.91%	0.09s
Faster R-CNN	47.59%	0.28s
MPNET(Faster R-CNN)	49.74%	0.26s
PVANET	75.52%	0.207s
MPNET(PVANET)	75.95%	0.191s

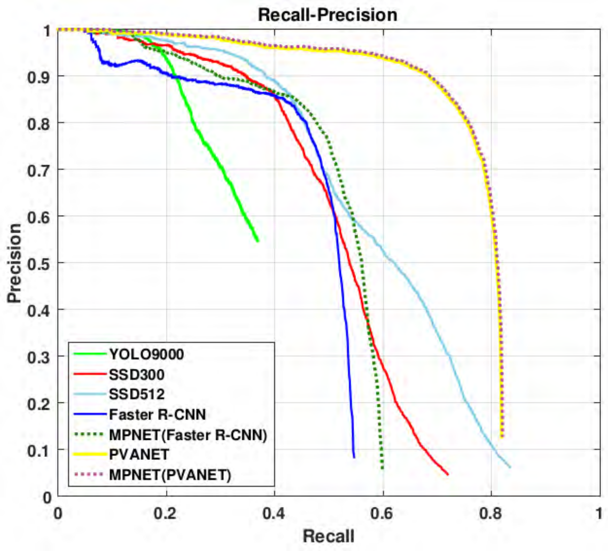

FIGURE 8. The recall-precision curve of pedestrian detection in XJTU dataset.

Table 2 and Fig. 8 show the performance on XJTU dataset. We can find that our method MPNET (Faster R-CNN) achieves 2.15% gain compared to Faster R-CNN and MPNET (PVANET) achieves 0.37% gain compared to PVANET. In addition, our MPNET (PVANET) method achieves 42.86% and 23.93% gains compared to YOLOv2 and SSD. In aspect of detection time, our MPNET (VGG16) achieves 0.02s decrease compared to Faster R-CNN and MPNET (PVANET) achieves 0.019s decrease compared to PVANET.

2) PERFORMANCE ON VEHICLE DETECTION

Our proposed framework is evaluated on the AVSS dataset. Table 3 and Fig. 9 show the overall experimental results on AVSS dataset. From Table 3, we can see that our framework outperforms other algorithms. In aspect of average precision, our MPNET (Faster R-CNN) method gains of 9.83% compared to the Faster R-CNN and MPNET (PVANET) gains of 2.34% compared to PVANET. In addition, our MPNET method achieves state-of-art performance. The detection time also decreases compared to region-based methods. For example, the detection time of MPNET (Faster R-CNN) decreases 0.02s compared to Faster R-CNN.

TABLE 3. Detection results on AVSS Dataset.

Methods	AP	Time
Yolo9000	39.39%	0.03s
FGFA	23.32%	0.40s
SSD300	62.69%	0.03s
SSD512	73.08%	0.09s
Faster R-CNN	41.17%	0.17s
MPNET (Faster R-CNN)	51.00%	0.15s
PVANET	86.46%	0.19s
MPNET(PVANET)	88.80%	0.16s

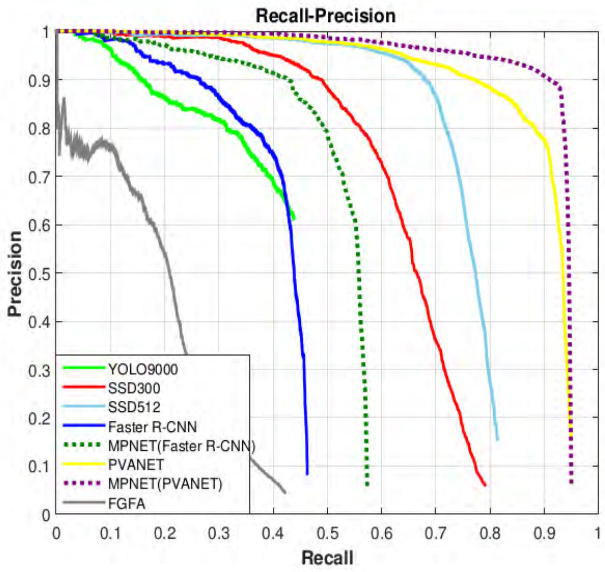

FIGURE 9. The recall-precision curve of vehicle detection in AVSS dataset.

We attempt to compare our method with FGFA [36], which also use spatial/temporal info. The average precision (AP) of FGFA is 23.32% with the time of 0.40s, which is 65.48% lower than MPNET (PVANET).

3) SUBJECTIVE PERFORMANCE

As shown in Fig. 10 and Fig. 11, we can find that the methods based on Deep Convolutional Neural Network are robust to detect the objects with high-resolution. However, they are not robust to the objects with low-resolution or occlusion. Our method is effective to solve these problems as shown in Fig. 10(d) and Fig.11 (d).

As shown in Fig. 12, the performance of our MPNET (Faster R-CNN) is better than the performance of Faster R-CNN. In addition, the performance of our MPNET (PVANET) is better than the performance of PVANET. Especially, the distant vehicles with low-resolution always miss in Faster R-CNN. However, our method performs well. In addition, the precision of the bounding boxes are greater than other methods. For example, the location of black car in Fig.12 (g) is more accurate than the location of red car in Fig.12 (e). By observing the detection results, we find that FGFA is not robust in some surveillance videos. This is due to the following reasons. Firstly, some objects are continuously missed in the process of detection due to the variation of detection results in adjacent (nearby) frames. In FGFA, the feature maps from nearby frames are warped to the reference frame according to the flow motion. Therefore, it could be a reason which results in continuous miss. In order to solve this problem, we fuse moving-object proposals generation and prediction because they are complementary to obtain most of objects. We propose the confidence-based prediction method to obtain stationary objects continuously and solve the fluctuation between adjacent frames. Secondly, due to camera angle and motion blur, there are some erroneous detection results. As shown in Fig. 12(b), the location of some bounding boxes are not accurate. In our method, our proposals generation framework is more accurate than the methods search on the whole frame, which can avoid generating unsuitable proposals. Finally, some distant objects are always missed due to the low-resolution of the distant objects. Therefore, the detection results don't have excellent performance. So we propose to enhance the detection results by fusing the moving-objects detection results and the predicted detection results, which strengthen the true positive and suppress the false positive.

V. DISCUSSIONS

A. IMPACT OF t_s

t_s is the threshold to distinguish high-confidence bounding box and low-confidence bounding box. The experiments in Table 4 show that $t_s = 0.5$ achieves the best performance. When t_s is greater than 0.5, AP decreases. This results from that unnecessary adjustment may cause inaccurate proposals. Therefore, when t_s is greater than 0.5, most of the boxes need to be adjusted, which is redundant and cannot achieve the best performance. When t_s is lower than 0.5, AP also decreases. Some bounding boxes with low-confidences are often with the problem of illuminance variation, occlusion, etc. Therefore, if we do not adjust the locations, these objects will not be detected accurately. We discuss the influence of t_s using MPNET (PVANET) in AVSS dataset.

TABLE 4. Impact of $\mathbf{t_s}$ in AVSS dataset.

t_s	0.1	0.3	0.5	0.7	0.9
AP	88.37%	88.49%	88.80%	88.49%	88.52%

B. IMPACT OF t_{min}

$t_{min}(t_{min} < t_s)$ is the lower limit of low-confidence bounding box and the threshold to suppress the false positive. We discuss the influence of t_{min} using MPNET (PVANET) in AVSS dataset. From Table 5, we find that $t_{min} = 0.3$ achieve the best performance. When $\mathrm{t_{min}}$ is greater than 0.3, the AP decreases. The reason is that some objects with low confidence are limited to predict and some bounding box are suppressed which may be the true positive. In other case, when $\mathrm{t_{min}}$ is lower than 0.3, AP decreases. This result from inaccurate proposals generated by unnecessary adjustment and invalid suppression.

TABLE 5. Impact of $\mathbf{t_{min}}$ in AVSS dataset.

t_{min}	0.1	0.2	0.3	0.4
AP	88.71 %	88.76%	88.80%	87.15 %

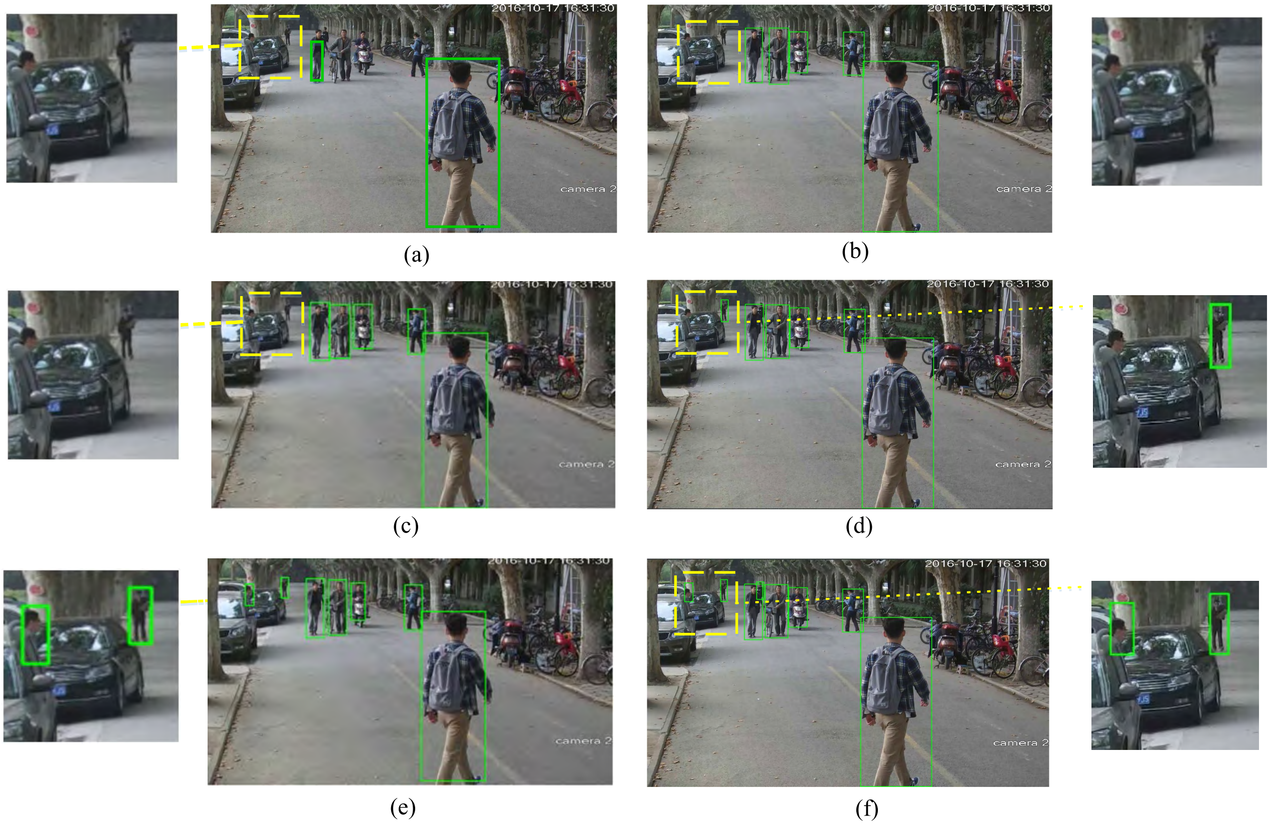

FIGURE 10. The comparison of four methods in XJTU Pedestrian dataset for pedestrian detection. (a) YOLO9000. (b) (b) SSD512. (c) (c) Faster R-CNN. (d) (d) PVANET. (e) MPNET(Faster R-CNN). (f) MPNET(PVANET).

C. IMPACT OF MOVING-OBJECT PROPOSALS GENERATION

Since our MPGP framework can be divided into moving-object proposals generation and confidence-based proposals prediction, we discuss the influence of the two parts, respectively. Our experiments evaluate in AVSS dataset. As shown in Table 6, if we only use the moving-object proposals generation to generate proposals (M-NET), the average precision of M-NET (Faster R-CNN) is only 38.98% while the average precision of M-NET (PVANET) is only 53.12%. This results from losing some stationary objects and the influence of fluctuations. When we add the prediction part, the AP increases 12.02% and 35.68%, respectively. This experiment shows that these two parts are inseparable. They can help each other to improve the performance.

TABLE 6. Impact of moving-object proposals generation.

Methods	AP
M-NET(Faster R-CNN)	38.98%
MPNET(Faster R-CNN)	51.00%
M-NET(PVANET)	53.12%
MPNET(PVANET)	88.80%

TABLE 7. Impact of confidence-based proposals prediction.

Methods	AP
Faster R-CNN	41.17%
Faster R-CNN+P	49.05%
PVANET	86.46%
PVANET+P	88.58%

D. IMPACT OF CONFIDENCE-BASED PROPOSALS PREDICTION

As mentioned above, our confidence-based proposals prediction method can solve the problem of confidence fluctuation. In order to improve the effectiveness of our confidence-based proposals prediction method, we combine Faster R-CNN and PVANET with our confidence-based proposals prediction method, which we named Faster R-CNN+P and PVANETX+P, respectively. Our experiments evaluate in AVSS dataset. The experiments show that we achieve 7.88% gain compared to Faster R-CNN and 2.12% gain compared to PVANET.

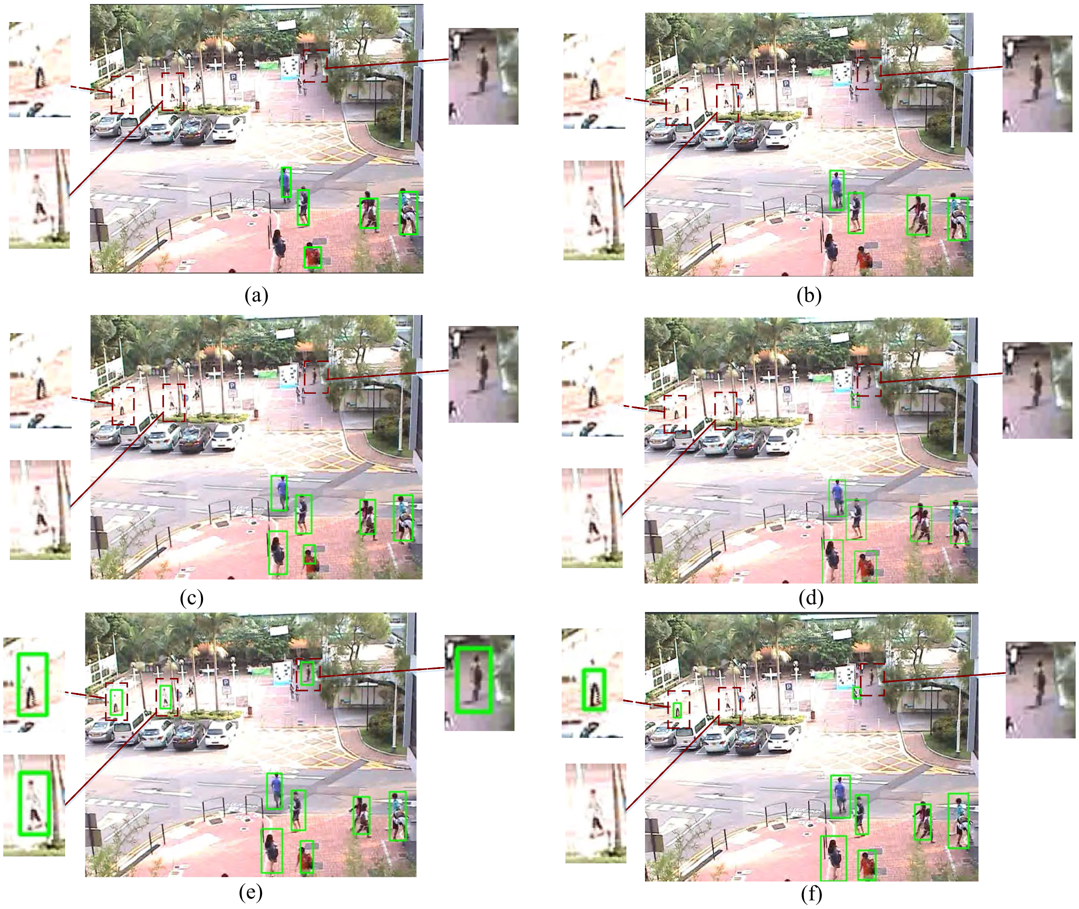

FIGURE 11. The comparison of four methods in CUHK Square dataset for pedestrian detection. (a) YOLO9000. (b) SSD512. (c) Faster R-CNN. (d) PVANET. (e) MPNET(Faster R-CNN). (f) MPNET(PVANET).

E. IMPACT OF MOTION-PROBED SEEDS ADJUSTMENT ANALYSIS

As mentioned above, we need to adjust the motion-probed seeds since the appearance and the speed of moving objects are diverse. We discuss the influence of adjustment analysis in AVSS dataset. From Table 8, we can find that the average precision of MPNET (Faster R-CNN) increases 1.15% compared to MPNET (Faster R-CNN) without adjustment analysis. In addition, average precision of MPNET (PVANET) increases 0.23% compared to MPNET (PVANET) without adjustment analysis.

TABLE 8. Impact of motion-probed seeds adjustment.

Methods	AP
MPNET (Faster R-CNN)-without adjustment	49.85%
MPNET (Faster R-CNN)	51.00%
MPNET (PVANET)-without adjustment	88.57%
MPNET (PVANET)	88.80%

F. IMPACT OF MULTI-SCALE NEURAL FEATURES

We discuss the influence of the multi-scale features in AVSS dataset. From Table 9, we can find that the average precision (AP) of single-scale is lower than the AP of multi-scale, which we use five different-size feature maps. We use Convk to represent the feature maps between the k-th pooling layer and the (k-1)th pooling layer to extract moving objects, which have the same size. The AP of MPET (Faster R-CNN) with multi-scale increased 1.23% compared to the AP of MPNET (Faster R-CNN) which only uses Conv1. The AP of MPET (PVANET) with multi-scale increased 3.43% compared to the AP of MPNET (PVANET) which

TABLE 9. The comparison of single-scale and multi-scale features.

Methods	MPNET (Faster R-CNN)	MPNET (PVANET)
Conv1	49.77%	85.37%
Conv2	45.14%	85.74%
Conv3	46.73%	86.00%
Conv4	42.46%	86.08%
Conv5	42.50%	86.08%
Conv1+2	49.83%	85.75%
Conv1+2+3	50.86%	86.11%
Conv1+2+3+4	50.92%	87.32%
Conv1+2+3+4+5	51.00%	88.80%

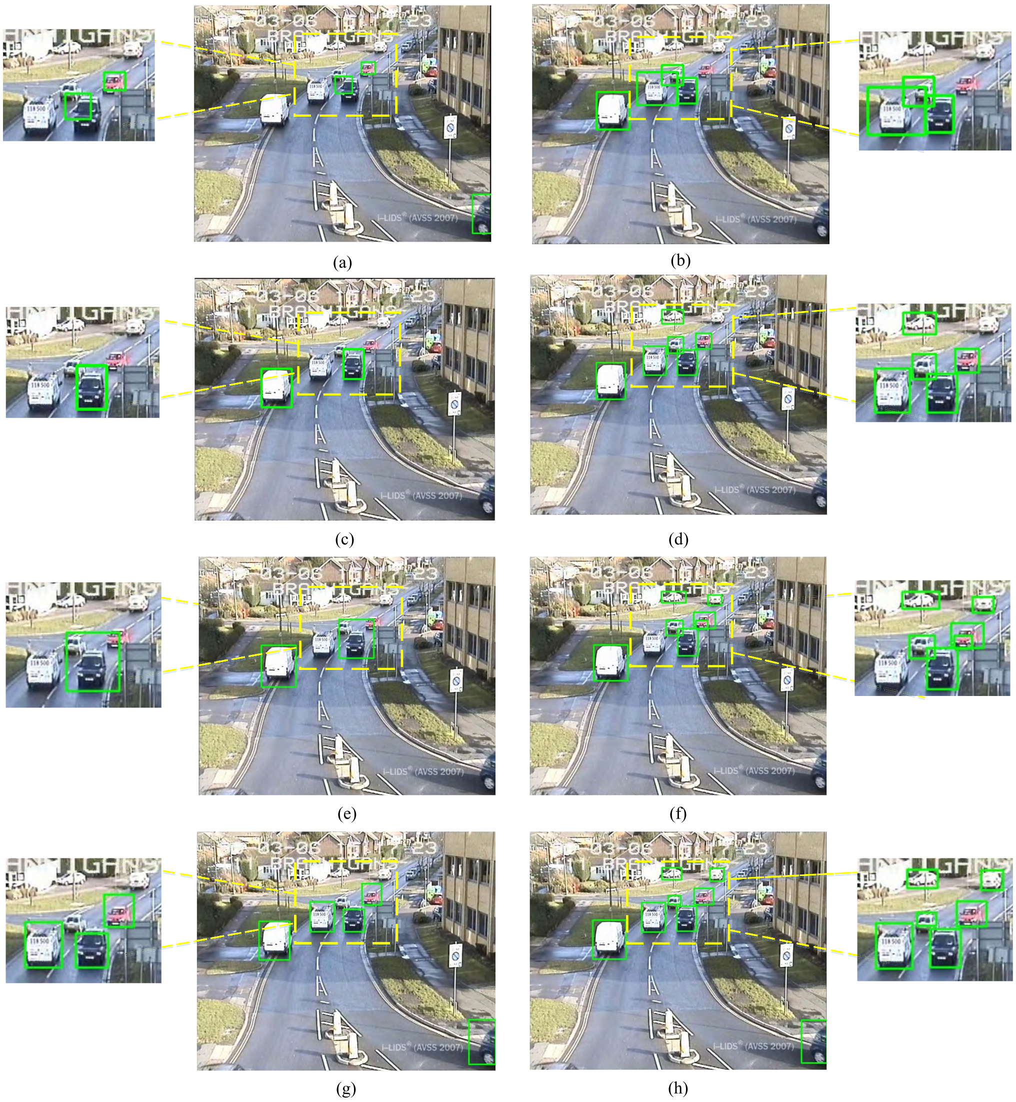

FIGURE 12. The comparison of our methods and other methods in AVSS Square dataset for vehicle detection. (a) YOLO9000. (b) FGFA. (c) SSD300. (d) SSD512. (e) Faster R-CNN. (f) PVANET. (g) MPNET(Faster R-CNN). (h) MPNET(PVANET).

only uses Conv1. In addition, we can see that the AP grows as the number of converged layers increases. Therefore, from Table 9, we can find that the frame differences over multi-scale neural features are better than the single scale counterpart.

VI. CONCLUSION

In this paper, we propose a Moving-object Proposals Generation and Prediction Framework (MPGP) to use spatial-temporal information to generate high-confidence proposals for object detection in surveillance videos. We explore the spatial-temporal in different convolutional layers to achieve accurate moving-object proposals. Experiments show that our moving-object detection proposals generation and confidence-based proposals prediction are complementary and all contribute to performance improvements. Only use one of the two parts cannot play the greatest role. In addition, we propose a proposals adjustment method, which is also effective to improve the detection results. We refine the results which contributes to the location precision of true positive and the suppression of false positives. Compared to traditional methods and other deep networks, our method shows superior performance in average-precision.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, Jun. 2014, pp. 580–587.

[2] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Dec. 2015, pp. 1440–1448

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[4] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. (2016). "PVANET: Deep but lightweight neural networks for real-time object detection." [Online]. Available: https://arxiv.org/abs/1608.08021

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. ECCV*, 2014, pp. 346–361.

[6] K. Kang, W. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutional neural networks," in *Proc. CVPR*, Jun. 2016, pp. 817–825.

[7] J. Redmon, S. K. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Jun. 2016, pp. 779–788.

[8] W. Liu *et al.*, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, vol. 1, 2016, pp. 21–37.

[9] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[10] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. ECCV*, vol. 5, 2014, pp. 391–405.

[11] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. CVPR*, Jun. 2016, pp. 845–853.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1106–1114.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014. [Online]. Available: https://arxiv.org/abs/1409.1556

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014, pp. 818–833.

[15] 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.

[16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, vol. 1, Jun. 2005, pp. 886–893.

[17] 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, 2010.

[18] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. CVPR*, Jul. 2017, pp. 6517–6525.

[19] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 1–9.

[20] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

[21] M. Wang, W. Li, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," in *Proc. CVPR*, Jun. 2012, pp. 3274–3281.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. CVPR*, Jul. 2017, pp. 936–944.

[23] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. CVPR*, Jun. 2015, pp. 1–9.

[24] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. CVPR*, Jul. 2017, pp. 5244–5252.

[25] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[26] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 8, pp. 1532–1545, Aug. 2014.

[27] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. CVPR*, Jul. 2017, pp. 3296–3297.

[28] "i-lids dataset for AVSS," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveill.*, London, U.K., 2007.

[29] Y. Xue and X. Qian, "Vehicle detection and pose estimation by probabilistic representation," in *Proc. ICIP*, Sep. 2017, pp. 3355–3359.

[30] K. Kang *et al.* (2016). "T-CNN: Tubelets with convolutional neural networks for object detection from videos." [Online]. Available: https://arxiv.org/abs/1604.02532

[31] K. Kang *et al.*, "Object detection in videos with tubelet proposal networks," in *Proc. CVPR*, Jul. 2017, pp. 889–897.

[32] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing objects while learning their appearance," in *Proc. ECCV*, 2010, pp. 452–466.

[33] A. Joulin, K. Tang, and L. Fei-Fei, "Efficient image and video co-localization with Frank-Wolfe algorithm," in *Proc. ECCV*, 2014, pp. 253–268.

[34] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proc. CVPR*, Jun. 2012, pp. 3282–3289.

[35] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[36] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. ICCV*, Oct. 2017, pp. 408–417.

[37] H. Shuai, Q. Liu, K. Zhang, J. Yang, and J. Deng, "Cascaded regional spatio-temporal feature-routing networks for video object detection," *IEEE Access*, vol. 6, pp. 3096–3106, 2018.

[38] S.-C. Huang and B.-H. Chen, "Highly accurate moving object detection in variable bit rate video-based traffic monitoring systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1920–1931, Dec. 2013.

[39] B.-H. Chen and S.-C. Huang, "An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 837–847, Apr. 2014.

[40] Y. Yuan, X. Liang, X. Wang, D.-Y. Yeung, and A. Gupta, "Temporal dynamic graph LSTM for action-driven video object detection," in *Proc. ICCV*, Oct. 2017, pp. 1819–1828.

[41] S.-C. Huang and B.-H. Do, "Radial basis function based neural network for motion detection in dynamic scenes," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 114–125, Jan. 2014.

[42] B.-H. Chen and S.-C. Huang, "Probabilistic neural networks based moving vehicles extraction algorithm for intelligent traffic surveillance systems," *Inf. Sci0*, vol. 299, pp. 283–295, Apr. 2015.

HANYU WANG received the B.S. degree from Xi'an University of Post and Telecommunications, Xi'an, China, in 2015, and the M.S. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, in 2018.

PING WANG received the B.S., M.S., and Ph.D. degrees from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, China, in 1999, 2002, and 2011, respectively. She was a Lecturer with Xi'an Jiaotong University from 2003 to 2014, where she is currently an Associate Professor. Her research interests include image processing, video coding, and video analysis.

XUEMING QIAN (M'10) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an, in 2008. He was a Visiting Scholar with Microsoft Research Asia from 2010 to 2011. He was an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014. He is currently a Full Professor with Xi'an Jiaotong University. He is also with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, with the Smiles Laboratory, Xi'an Jiaotong University, and ZhiBiAn Technology (as a Consultant). His research is supported by the National Natural Science Foundation of China, Microsoft Research, and the Ministry of Science and Technology. His research interests include social media big data mining and search. He received the Microsoft Fellowship in 2006. He also received outstanding doctoral dissertations of Xi'an Jiaotong University and Shanxi Province, in 2010 and 2011, respectively.