IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE (AI)-EMPOWERED INTELLIGENT TRANSPORTATION SYSTEMS

Received February 26, 2020, accepted March 15, 2020, date of publication March 18, 2020, date of current version March 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981821

P-LPN: Towards Real Time Pedestrian Location Perception in Complex Driving Scenes

YI ZHAO[1,3], (Member, IEEE), MINGYUAN QI[2], XIAOHUI LI[1], YUN MENG[1], (Member, IEEE), YAXIN YU[1], (Senior Member, IEEE), AND YUAN DONG[1]

[1]School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China
[2]School of Information Engineering, Chang'an University, Xi'an 710064, China
[3]JYI Intelligent-Tech Group, Shaoxing 312000, China

Corresponding author: Yi Zhao (z1@chd.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701044 and Grant 61704010, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM5165, in part by the Shaanxi Key Project of Research and Development Plan under Grant 2019ZDLGY03-01, and in part by the Xi'an Science and Technology Project under Grant 201805045YD23CG29.

ABSTRACT Semantic segmentation is one of the most critical modules in road scene understanding. In this paper, we focus on the challenging task of pedestrian's relative location perception in the semantic graph of complex driving scenes. Prevalent research on semantic segmentation mainly concentrate on improving the segmentation accuracy with less attention paid to computational efficiency. Furthermore, little effort has been made in pedestrian location perception in complex driving scenes. For example, current semantic segmentation methods classify all pedestrians as a mono category, regardless of whether the pedestrians are penetrating into the vehicular lane or standing still in the safe sidewalk area. We propose a pedestrian location perception network (P-LPN). P-LPN can produce real-time semantic segmentation while simultaneously providing location inference for each pedestrian in semantic maps. This enables autonomous driving system to categorize pedestrians into different safety levels. We comprehensively evaluated P-LPN on CityScapes benchmark through comparative studies. Our proposal achieved competitive performance in both accuracy and efficiency. It yields quality inference with real-time speed at ∼22 fps.

INDEX TERMS Semantic segmentation, deep learning, road-scene understanding, pedestrian perception.

I. INTRODUCTION

Pedestrian-vehicle accidents highly likely result in incapacitating injuries and fatalities. The death toll for pedestrians has risen sharply during the past decade globally. According to Insurance Institute for Highway Safety (IIHS) 2018 report [1], pedestrian fatalities increased 46% from 2009 to 2016 in USA. In China, World Health Organization(WHO) has estimated that over 60% of traffic-related fatality each year were vulnerable road users including pedestrians, cyclists, etc., [2]. Therefore, for any autonomous driving system, it should improve the safety of different road user groups, non-motorized road users particularly. Indeed, the intelligent vehicle should be aware of the relative location and dynamic behavior of these specific traffic participants. This has always been one of the major challenges for autonomous driving. Among all vehicle-mounted sensors (Visible-spectrum cameras, LIDAR, MMWR and ultrasound radars), On-car cameras are most indispensable. It is less expensive and capable of providing real time vision for surrounding perception and road scene understanding [3].

Semantic segmentation is currently the most prominent machine vision technique in traffic scene understanding [4]–[7]. The aim of semantic segmentation is to perform accurate pixel-wise classification on the image, parsing it into different semantic categories. For example, in a driving scene, images are usually parsed into pedestrians, roads, lanes, curbs, sidewalk, traffic signals and buildings, etc (as shown in Fig.1). Recently, deep learning based methods have shown a rapid growth in autonomous driving systems and Intelligent transportation system. Typical applications include deep reinforcement learning in vehicular networks and vehicular edge computing [8], [9], road feature extractions [10], etc. Noteworthily,with large scale training data source [5], high

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

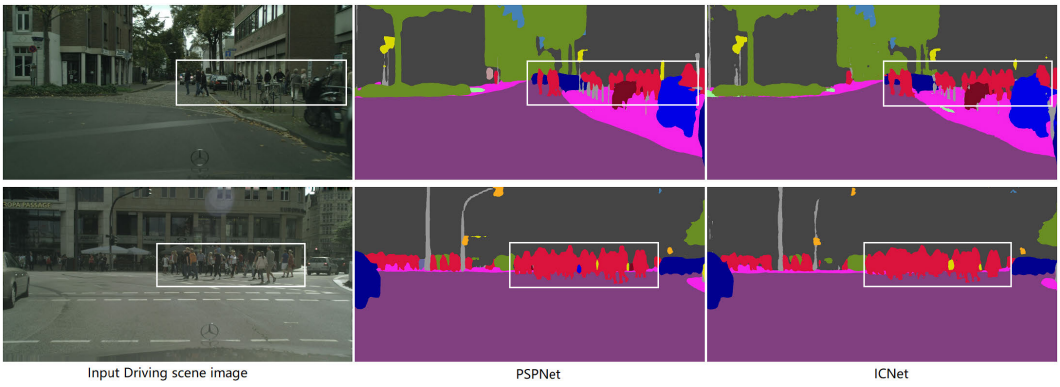


FIGURE 1. State-of-art semantic segmentation methods of PSPNet and ICNet in pedestrian perception.

performance computing hardware (GPUs, TPUs) and solid deep learning frameworks [12]–[14], deep learning based semantic segmentation methods have achieving tremendous progress and becoming dominant solutions by outperforming other state-of-art approaches. The major interest is that they can provide end-to-end framework in classifying objects at the pixel level [11]. Badrinarayanan *et al.* [15] proposed an encoder-decoder architecture termed SegNet. SegNet is a full convolutional neural network (FCNN) based semantic segmentation model. It employs decoder network to up-sample the feature maps by retaining the max-pooling indices from the corresponding encoder layer. Zhao *et al.* studied a pyramid pooling based scene parsing model PSPNet [16]which further improves the segmentation accuracy. Chen pioneered the use of Atrous convolutions and full connected Conditional Random Field (CRF) in a series of effective semantic segmentation solutions (DeeplabV1 to DeeplabV3) [17], [18]. Zhang *et al.* proposed a more efficient asymmetric encoder-decoder structure for semantic segmentation, it has much fewer parameters than Segnet [22]. In fact, previous deep learning based semantic segmentation models suffer from low efficiency, they mainly exploit fully convolutional networks (FCNs) which is a sophisticated architecture with multiple layers of convolution, pooling, and normalization, etc. For pixel-level prediction tasks like semantic segmentation, improving accuracy means increasing model complexity and number of operations. This will in-return sacrifice the efficiency. Therefore, in recent years, some efforts have been made to improve the efficiency of semantic segmentation models. Zhao *et al.* proposed a cascade image input structure (ICNet) which obtains much higher speed [24]. Romera *et al.* used residual factorized convolutions to optimize the efficiency of the segmentation model [19]. Siam *et al.* presented a real-time segmentation benchmarking framework for quick prototyping of different encoder-decoder architecture [23]. In terms of pedestrian intention estimations, Keller studied different stereo-video based pedestrian path prediction methods [20]. Latter, Fang proposed a pedestrian crossing-stoping intention classification method using deep learning based 2D pose estimation [21].

Although Deep learning along with machine vision techniques has made remarkable progress on driving scene understanding and pedestrian perception, yet, a simple but fundamental problem remains unsolved: How to parse the pedestrian's relative location in a complex driving scene image? In fact, the relative locations of pedestrians are directly related to their safety. A pedestrian abruptly rushing into driving lane may result in severe incident for both himself and the autonomous vehicle. Contrarily, pedestrians walking on the sidewalk pavement are away from road area, they present no threat to driving safety and can hardly interfere with the traffic. As introduced above, previous works did not focus on this. For example, semantic segmentation models mostly focus on improving the segmentation accuracy with almost no attention paid to parse further pedestrian location related semantic information. As shown in Fig. 1, these methods regard all pedestrians as a mono category no matter whether the pedestrian is crossing the driving lane or walking in the sidewalk area. All pedestrians are marked with the same color and the same label.

To address this problem, we proposed an efficient fine-grained Pedestrian Location Perception Network abbreviated as 'P-LPN'. It is backboned with our proposed high efficient semantic segmentation model termed Inner Cascade Network (InCNet) along with an adjusted region proposal network (RPN) and a location perception layer (LP-layer). The proposed deep learning model can effectively distinguish pedestrian's relative location in a complex driving scene. Several improvements have been made to the architecture so that P-LPN can reach the balance between inference accuracy and efficiency.

Our main contributions are the following.

1) We proposed a high efficient and low computation cost semantic segmentation model termed Inner Cascade Network (InCNet) for real-time semantic segmentation. With the inner cascade structure and cascade feature fusion, it enables fusion and refining low-resolution semantic graph with details from high resolution images.

2) Based on our InCNet, we develop a novel pedestrian location perception network named P-LPN. It mainly combines the InCNet and a Region proposal network (RPN). P-LPN is the first end-to-end model achieving real-time driving scene parsing and pedestrian locations perception at the same time. Experiments on the latest driving-scene parsing benchmark of CityScapes proved its competitive performance in both accuracy and efficiency.

We made P-LPN publicly available.[1] A detailed explanation of P-LPN is given in Section 2. Experiments and results are presented in Section 3 and Section 4 respectively. Section 5 concludes this paper.

II. PEDESTRIAN SEMANTIC PERCEPTION NETWORK (P-LPN)

Through efficient scene parsing and region proposal network, P-LPN can provide real-time fine-grained pedestrian location perception, identifying different type of pedestrian's relative locations: in driving area(driving lanes, driving pavement) or out of driving area (Sidewalk, safe-island, etc). With P-LPN, an autonomous driving system can distinguish those who invading the near-front driving lane and those who staying in sidewalk area. In this section, detailed description on P-LPN is given.

As shown in Fig. 2, P-LPN mainly consists of three modules: first, our proposed semantic segmentation model of Inner Cascade Network (InCNet) is employed for fast scene parsing. Secondly, an adjusted Region Proposal Network (RPN) is used to further propose several possible bounding box for each pedestrian in the feature maps. In module 3 (LP layer), we merge the results from module1 and module2, it identifies the relative location of each pedestrian (In driving lane or in sidewalk area, etc.) and marked them with different colored bounding box and labels with location information.

A. THE INNER CASCADE NETWORK (InCNet) BACKBONE IN P-LPN

Inspiring by the state-of-art semantic segmentation model Image Cascade Network (ICNet [24]), we adjusted the model by proposing a more practical semantic segmentation framework named Inner Cascade Network (InCNet).

As explained in introduction, previous semantic segmentation methods usually exploit full convolutional neural network (FCNN) with unique-sized input. It can be very time-consuming in parsing high-resolution images. Contrary to previous methods, ICNet employs a cascade architecture for semantic segmentation. It adopts cascade image inputs (i.e., low-, medium- and high resolution images), computational-intensive task like generating coarse prediction maps only go through low-resolution branch. Then, cascade feature fusion (CFF) unit strategy helps to merge the output of coarse semantic graph with medium and higher resolution branches. This serves to fine-tune the coarse semantic map gradually. Therefore, the most computation expensive task was carried out on lowest resolution input while the high resolution input only go through light-weighted CNN. As a result, the whole computation costs significantly reduced.

To further improve the practicality of ICNet, we proposed InCNet which is illustrated in the middle part of Fig. 2. The main difference between our proposed InCNet and ICNet is that our proposed InCNet has an inner cascade architecture, it only has one input image with full resolution. we eliminate the down sampling operation to form half size image and quarter size image in the input stage. Therefore, Before the inner cascade structure, the weights and computations of stage 1 of resnet 50 core is completely shared. Thus, InCNet has only one image input, also, it saved the input image down sampling computations and the operation on three independent convolutional layers.

InCNet works as follows: At the fist stage, the input image is with full resolution (e.g., 1024 $\times$ 2048 in CityScapes), it is fed directly into a full semantic perception network (resnet50 cored [30]), after the convolution layer of stage 2 in resnet 50. a 1/8 sized feature map is taken out(high resolution branch). Then, the medium resolution branch (1/16) get out after the convolution layer of stage 3 in the resnet 50. At this stage, previous layers of resnet 50 are all with shared weights and computations. the rest 1/16 feature maps get through the whole layers and come out with a 1/32 feature map. At the second stage, the output feature maps are fused by CFF unit from low to high resolution. The medium and higher resolution branch together can help to fine-tune the coarse prediction map generated by the low-resolution branch. The model are trained with cascade label guidance strategy [24] in which different-scale (1/16, 1/8, and 1/4) ground-truth labels are used in the learning stage for the three branches respectively. With this inner cascade structure, The InCNet is highly efficient and memory saving.

At the output of InCNet, we made an adjustment in order to obtain higher speed for the following operations. Unlike ICNet, in the up-sampling stage, we only kept the $\times 2$ up-sampling while the final $\times 4$ up-sampling operation was abandoned. This means that the output of our InCNet is the 1/4 size of the original full resolution image. Thus, the following operation like filtering and searching are all conducted on the 1/4 size semantic graph which helps to further reduce the computational costs.

B. ADJUSTED REGION PROPOSALS NETWORK(RPN)

As discussed in introduction, current semantic segmentation methods take all pedestrians as a mono category, therefore, a group of crowded pedestrians are usually marked as a glob of indistinguishable pixels in the semantic graph. This is inadequate for driving scene understanding. In this work, we further adjusted the Region Proposal Network (RPN) [25], [26] to locate each pedestrian's position on the semantic fields produced by the InCNet.

Generally, a RPN can be understood as an attention model. It is a light weighted network taking an image (of any size) as

[1] www.github.com/espci

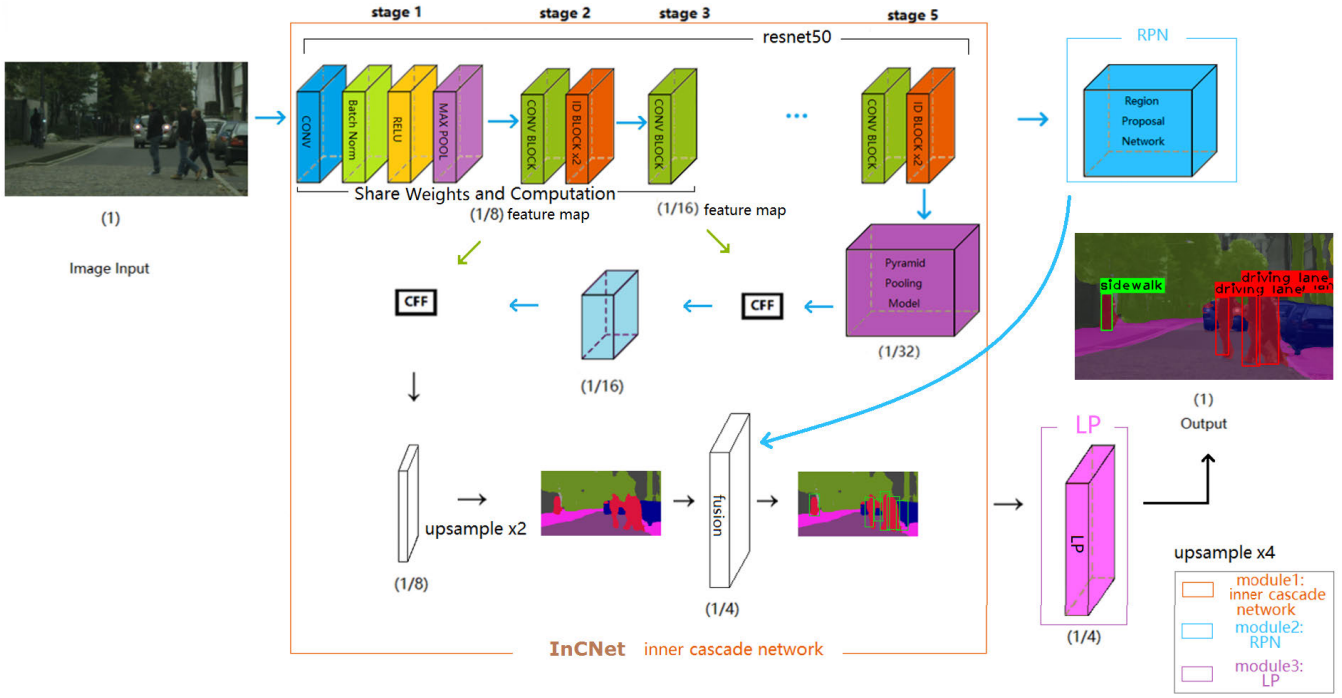


FIGURE 2. Overall architecture of P-LPN.

input and generating outputs of rectangular object proposals, each with an objectness score. It consists of three stages: Basic convolutional layers transfer original image to feature map. Then, it is followed by a small network which takes as input a n × n spatial window from the feature map (n=3 in this paper). Each small network will slide over the feature map and obtain lower dimensional feature. The outputs of this small network will be fed into two smaller sliding fully-connected layers (1 × 1 convolutional layer in this work): a box-regression layer (reg) and a box-classification layer (cls). During the generation of region proposals, for each sliding window location, multiple region proposals are predicted. The number of most likely proposal for each location is denoted as k. Therefore, the reg layer has 4 k outputs for the coordinates of k boxes, while the cls layer has 2k outputs that estimates the classification probability for the object in each proposal. Indeed, each sliding window is correspondent to a reference box called 'anchor'. The anchor is located at the center of each proposal with different scales and aspect ratios. By default, there are three scale and three aspect ratio for the anchor, which makes 9 anchors at each sliding position. Finally, the output of RPN contains a set of anchor boxes correspondent to the proposals.

In this work, we made two major modifications on RPN (Fig. 3.) to make it more cost-efficient and more adaptable for P-LPN. First, to reduce the computation costs of RPN, sharing computation with the InCNet (Module1 shown in Fig. 2.) is probably the best option. Therefore, in our adjusted RPN, the input convolutional layers in RPN are totally eliminated, we take the output feature map of Resnet50 core from the

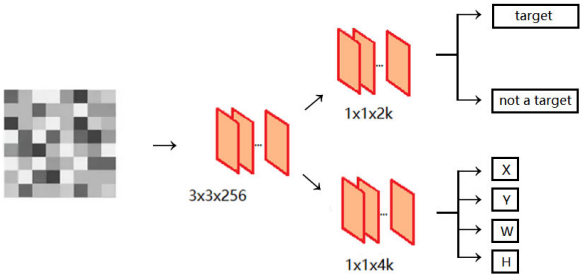


FIGURE 3. Region proposals network.

upper branch of InCNet directly into the input of RPN (up-right in Fig. 2.). This can significantly reduce the computational costs by sharing the whole Resnet50 with the InCNet and removing the RPN's own convolutional layers.

Secondly, in a typical region proposal network, there are 9 anchors (3 scale × 3 aspect ratio) at each sliding position. Thus, for a convolutional feature map of a size $W \times H$, there are $9 \times W \times H$ anchors in total. However, for P-LPN, our attention mainly focuses on pedestrians. As most pedestrians can fit well with vertical rectangle boxes. Therefore, we kept only the vertical rectangle aspect ratio with five different scales for pedestrians at different distance (5 scale × 1 aspect ratio). This makes $5 \times W \times H$ anchors for a convolutional feature map of size $W \times H$, saving 45% of the computation cost.

C. THE LOCATION PERCEPTION LP-LAYER

LP layer is the last module in P-LPN, it is the abbreviation of location perception. As shown in the right bottom

of Fig.2. After we fused the results from RPN and the InCNet, we obtain a semantic graph with several region proposal boxes. As for the LP layer, the objective is to filter the impropriate or irrelative bounding boxes and identify the relative location of the each pedestrian. A detailed explanation of these two operations are given below.

1) BOUNDING BOX FILTERING

The input of LP contains several bounding boxes (proposal) of pedestrians, some are perfectly suitable while some are inaccurate. So we need to keep the appropriate bounding boxes while eliminate the incorrect ones. This operation can be done by calculating the occupation percentage of pedestrian in the bounding box. The bounding boxes that properly frame the pedestrians have a higher occupation ratio while impropriate bounding boxes have much lower occupation ratio. A classical way to calculate the occupation ratio is calculated pixel by pixel with 2 nested loops illustrated in the following Method I.

Method I: Pedestrian Occupation Ratio Calculation With Nested Loops (on CPU)

Input: A cropped pedestrian image
Output: Percentage of pedestrian in an image
Begin
  for i in range(image height)
    for j in range(image width)
      if label[i][j] = = pedestrian
        Count++
      end
    end
  end
  Percentage = Count/image size
  Return Percentage
End

Where the i,j refers to the pixel's coordinates. The label value is the predicted pixel's class by InCNet. We find this method is quite time consuming. In this work, we proposed a more efficient occupation ratio calculation method using average pooling operation on GPU. The proposed Method II is shown as following.

Method II (Proposed): Bounding Box Filtering With Average Pooling (on GPU)

Input: A pedestrian bounding box
Output: Percentage of pedestrian in an image
Begin
  Image = tf.image.crop_and_resize()
  Image − = tf.equal(Image,pedestrian label)
  $Percentage = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} pixel$
    (tf.avg_pool(Image))
  Return Percentage
End

Benefiting from Tensorflow-GPU'S high performance in parallel computing [27]. Our proposed method gains over

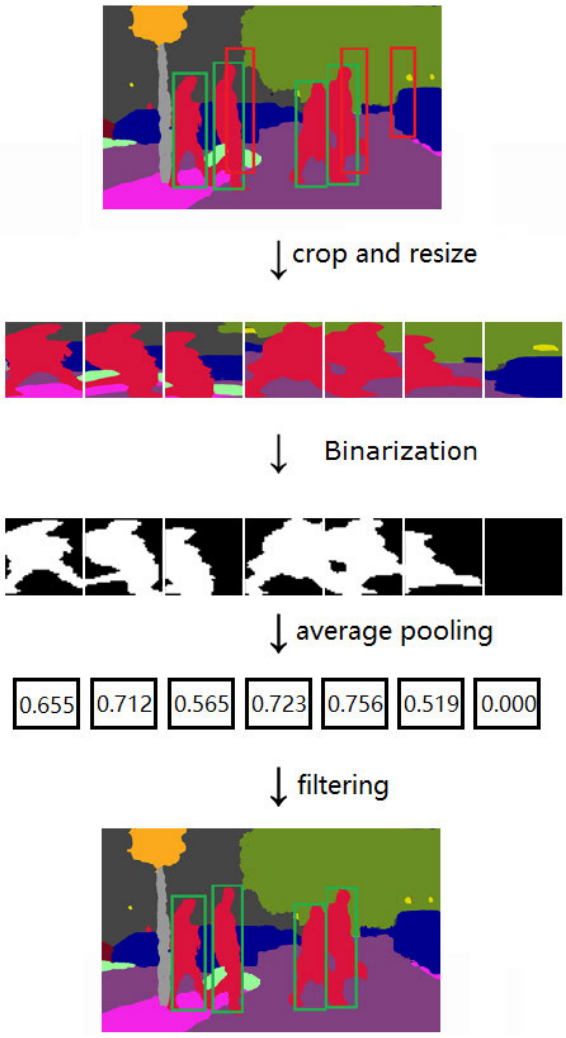


FIGURE 4. Bounding box filtering.

10× efficiency than pixel-wise calculation(Method I) on CPU. The test result suggests that the average processing time of Method II on GPU (Nvdia 1080Ti) costs 7.3ms while pixel-wise calculation (Method II) on CPU (Intel I5) takes more than 108.4 ms.

Further explanation of this novel method is exhibited in Fig.4. We first crop the bounding box on the semantic graph. Then, they are rescaled to the same size (e.g.14 × 14). Secondly, we extract pedestrians from each graph, labeling the pedestrian pixels as 1 while others as 0. Later, we fed the segmented binary frame into the average pooling layer to get the pedestrian occupation ratio of each frame. Finally, we eliminate the bounding box with lower scores and optimize the bounding box with (non maximum suppression) NMS algorithm.

2) PEDESTRIAN LOCATION PERCEPTION

After the previous operations, each pedestrian is framed with an appropriate bounding box, we further exploit a novel method to determine the relative location of each pedestrian

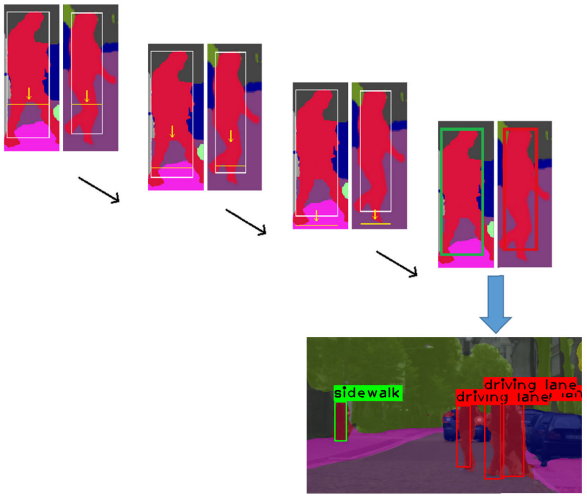


FIGURE 5. Pedestrian location identification on semantic graph.

in the semantic graph (Fig.5.). Firstly, we select a line near the bottom of the pedestrian bounding box (1/4 to the bottom) and move it downwards to the lowest point of the pixel labeled pedestrian (the lowest pixel on the feet of pedestrian, could be in or out of the bounding box). Below this line, the label of pixels on the semantic graph directly represents the semantic attribute of the standing area (driving lane, sidewalk pavement, safe-island, etc.). Finally, the location of this pedestrian can be determined according to the semantic label of his standing area. The pedestrians interfering with traffic flow (pedestrian standing or walking in the driving lane) will be tagged with labeled bounding boxes in red while the pedestrians out of the driving area (sidewalk or safety island) are marked with green bounding boxes. Although this method is simple, it is high efficient with high accuracy.

III. MODEL TRAINING AND EXPERIMENTAL SETTINGS

A. THE TRAINING OF P-LPN

There are two trainable deep learning modules in a P-LPN: the InCNet backbone and the adjusted RPN. The input of RPN is the output feature map of Resnet50 core from the InCNet backbone. Therefore, the training are separated in two phases where the two modules are trained separately, we set the learning rate of RPN to zero where only the InCNet are trained, then, we keep the parameters of the shared convolutional network (the Resnet50 core) in InCNet unchanged, in this phase the RPN are trained alone. We applied samilar training strategy and loss functions referring to previous works of [24]and [25] for InCNet and RPN respectively. For the InCNet, weighted softmax cross-entropy loss is adopted in each branch (Three inner branches of Small, Medium, High resolution), and the loss function is defined mathematically below:

$$L_{InC} = -\sum_{t=1}^{\tau} \lambda_t \frac{1}{Q} \sum_{y=1}^{\mathcal{Y}_t} \sum_{x=1}^{\mathcal{X}_t} \log \frac{e^{\mathcal{F}^t_{\hat{n},q_{(x,y)}}}}{\sum_{n=1}^{\mathcal{N}'} e^{\mathcal{F}^t_{n,q_{(x,y)}}}}. \quad (1)$$

where λ_t is related loss weight. τ is the total number of inner branch t, in the case of InCNet, we have $\tau = 3$. Q represents the spatial size ($x \times y$) of predicted semantic map. $q_{(x,y)}$ stands for a single pixel at position (x, y). Thus, the predicted per-pixel value in the semantic graph is $\mathcal{F}^t_{n,Q_{(x,y)}}$ while the correspondent ground truth label value is $\mathcal{F}^t_{\hat{n},Q(x,y)}$.

The loss function for training RPN consists of two parts: first, the $L_{cls}(Ac_i)$ is a classification loss of anchor i, it is a log loss over two classes (object or not object in the anchor). L_{reg} represents the location regression loss (smooth L_1) defined in [29]. $Ac^*_i L_{reg}(l_i)$ means $L_r eg$ loss only consider positive anchors. The loss function for RPN is expressed in the equation below:

$$L_{RPN}(Ac_i, l_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(Ac_i) + \lambda \frac{1}{N_{reg}} \sum_i Ac^*_i L_{reg}(l_i) \quad (2)$$

where the two losses are normalized by the mini-batch size N_{cls} and N_{reg}, λ is the balancing parameter. When the two modules are all well trained, the modules in P-LPN will be trained together with an overall loss function $\mathcal{L}$ which is minimized in the following expression below.

$$\mathcal{L} = \lambda_1 L_{InC} + \lambda_2 L_{RPN} \quad (3)$$

where the overall loss $\mathcal{L}$ is the weighted sum of losses of InCNet (L_{InC}) and RPN (L_{RPN}). λ_1 and λ_2 are the balancing weight parameters. After the P-LPN are well trained, it becomes end-to-end operational, providing quality semantic prediction and pedestrian location inference. In this work, the model training is with the open source cityscapes dataset [5]. During training, we randomly initialize the network's weights from a zero-mean Gaussian distribution with standard deviation 0.01. The InCNet Training is performed using a batch size of 8, number of iteration of 60000 times and the momentum optimization of Stochastic Gradient Descent. We used poly learning rate policy with the learning rate of 5e-4, a weights decay parameter of 1e-4, momentum parameter of 0.9. The RPN training is performed using a batch size of 2, 90 epochs and the Adam optimization of Stochastic Gradient Descent with a starting learning rate of 1e-3. On each epoch the learning rate adaptively adjusted.

B. EXPERIMENT SETTINGS

The whole framework was implemented with Tensorflow 1.12 on a GPU of NVidia GTX1080Ti. The model training and testing are conducted under the same hardware environment. The whole project is also publicly available in the footnote of page 2.

To further validate our proposal of P-LPN, we tested our model with 500 images from CityScapes test dataset. As objective of P-LPN is different to prevalent semantic segmentation models and there are lot of customized optimizations on P-LPN. Therefore, we find it difficult to conduct a systematic comparative study with all previous methods. However, we managed to rebuild another version

of P-LPN (P-LPN^{psp}) based on the state-of-art semantic segmentation model of PSPNet by replacing our inner InCNet with a PSPNet50 (Resnet50 cored). We also compared our model with the prevalent state-of-art methods in part. The experimental results and details of the comparative study are given in the following section. In this section, we investigated the P-LPN's performance through experiments on CityScapes dataset and comparative studies.

IV. RESULTS AND DISCUSSION

In this section, we investigated the P-LPN's performance through experiments and comparative studies.

A. PERFORMANCE EVALUATION INDICATORS

As explained in the previous section that P-LPN is a new framework for pedestrian relative location identification in driving scene, it is neither a semantic segmentation model (e.g. PSPNet [16], SegNet [22], ICNet [24], etc.) nor a target detection model (e.g. Faster RCNN [25], Mask RCNN [26], etc.). In fact, P-LPN can be considered as a hybridization of these two models. Therefore, it is difficult to find a systematic way to compare our work with existing ones mainly because there is no similar works before. Furthermore, the evaluation indicator for segmentation models and target detection models are totally different. For example, most segmentation models are evaluated by mean intersection over union (mIoU) and speed while object detection models use mean average precision (mAP), recall, bounding box mIoU (different to the mIoU in segmentation models), speed, etc. We cannot use all of these indicators to evaluate the performance of our proposal. They are calculated differently and some are insignificant for P-LPN. Therefore, we proposed six major indicators for P-LPN which are Average Precision (AP) for pedestrian detection performance, recall for evaluating missed detections,bounding box mIoU to evaluate whether the box properly framed pedestrian, time and FPS(Frame Per Second) for evaluating the speed of the model. Finally, Location Recognition Accuracy (LRA) for evaluating the location perception accuracy which is calculated by equation (4) below.

$$LRA = \frac{L_d + L_{od}}{N_p}. \tag{4}$$

where L_d and L_{od} refer to correct location recognitions of pedestrians in driving zone (driving lanes) and out of driving zone (sidewalk, safe-island, etc) respectively. N_p represents the total number of detected pedestrians.

B. EXPERIMENT RESULTS AND DISCUSSION

We comprehensively evaluated our proposal on the CityScapes benchmark [5]. We verified the performance of P-LPN on 500 images (1024 $\times$2048 sized) of complex urban driving scenes. With the same test data, we also conducted a comparative study of our proposal(P-LPN^{inc}) with state-of-art semantic segmentation model of PSPNet and ICNet. As shown in Fig. 6, our P-LPN made descent predictions results, most semantically meaningful objects have been correctly captured and segmented at pixel-level. Furthermore, It is noteworthy that compared to classic semantic segmentation models (PSPNet, ICNet), P-LPN is the first to make fine-grained location perceptions of pedestrians in complex driving scene. It effectively extracted each pedestrian in the semantic map. Better still, it accurately identified the relative position of each pedestrian. Those penetrating into the driving lane are framed with red bounding boxes and those standing in the sidewalk or safe-island areas are marked with green bounding boxes. Their relative location information are also marked in the labels. In Fig. 7, we compared P-LPN with state of art instance segmentation and object detection methods of Mask-RCNN(Tensorflow version) [26] and Yolo V3 [28]. As shown in Fig. 7, Yolo V3 and Mask R-CNN have detected objects with recognition bounding box and segmentation masks respectively. As shown in Fig. 7, Yolo V3 and Mask R-CNN marked the detected objects with recognition bounding box and segmentation masks respectively. Different from these two methods, P-LPN produced pixel-level semantic prediction while simultaneously providing pedestrian detection with the location inference.

The final comparative results with previous works and our proposal of InCNet cored P-LPN is given in Tab. 1. It suggests that both P-LPN^{Inc} and P-LPN^{psp} have a high Average Precision score, it is mainly because that for a P-LPN, the image is fed into a serial deep learning modules (two stage), the Front-end Resnet50 (PSPNet or InCNet), and the back-end RPN. This helps to achieve higher recognition accuracy. It is notable that target detection models like Mask RCNN and Yolo V3 are more sensitive to a person's features, which leads to a undesirable result: Some non-pedestrian targets have been identified as pedestrian in the driving scene, for example, the characters printed on the bus (row3 in Fig. 7) are wrongly identified as real pedestrians. This reduced their precision in general. The recall of pedestrian detection for both P-LPN models are lower, it is mainly because the adjusted RPN has a limited ability in small object detection. So the pedestrian far away from the vehicle can be very tiny in the vision field, these pedestrians are highly likely missed. Also, when a large part of a pedestrian is shielded behind other objects, it can be hard for RPN to detect this person. In terms of location recognition, all previous methods do not provide this function, only P-LPN can give the relative location perception on each pedestrian. This indicates that our LP layer can effectively identify the area of pedestrians. Admittedly, our proposed model P-LPN (InCNet core) sacrificed bounding box IoU for higher speed. Some IoU related operations with high computational costs are optimized in RPN and LS layer for higher speed. Actually, the bounding box IoU under this application scenario is much less significant than Recall and speed, so we think it is acceptable. It is notable that our proposed model outperformed all five investigated methods in key indicators of pedestrian recognition average precision (AP) and pedestrian location recognition accuracy (LRA), also, it ranks the second in the efficiency indicator of

FIGURE 6. Comparison of P-LPN with semantic segmentation models (PSPNet (upper image) and ICNet (under image) in the middle colomn).

time and fps: our proposal is 22× times faster than PSPNet, 102.7× times faster than Mask RCNN and 22.4 × times faster than PSPNet cored P-LPN. As speed is one of the most important factors in driving scene parsing, we believe that our

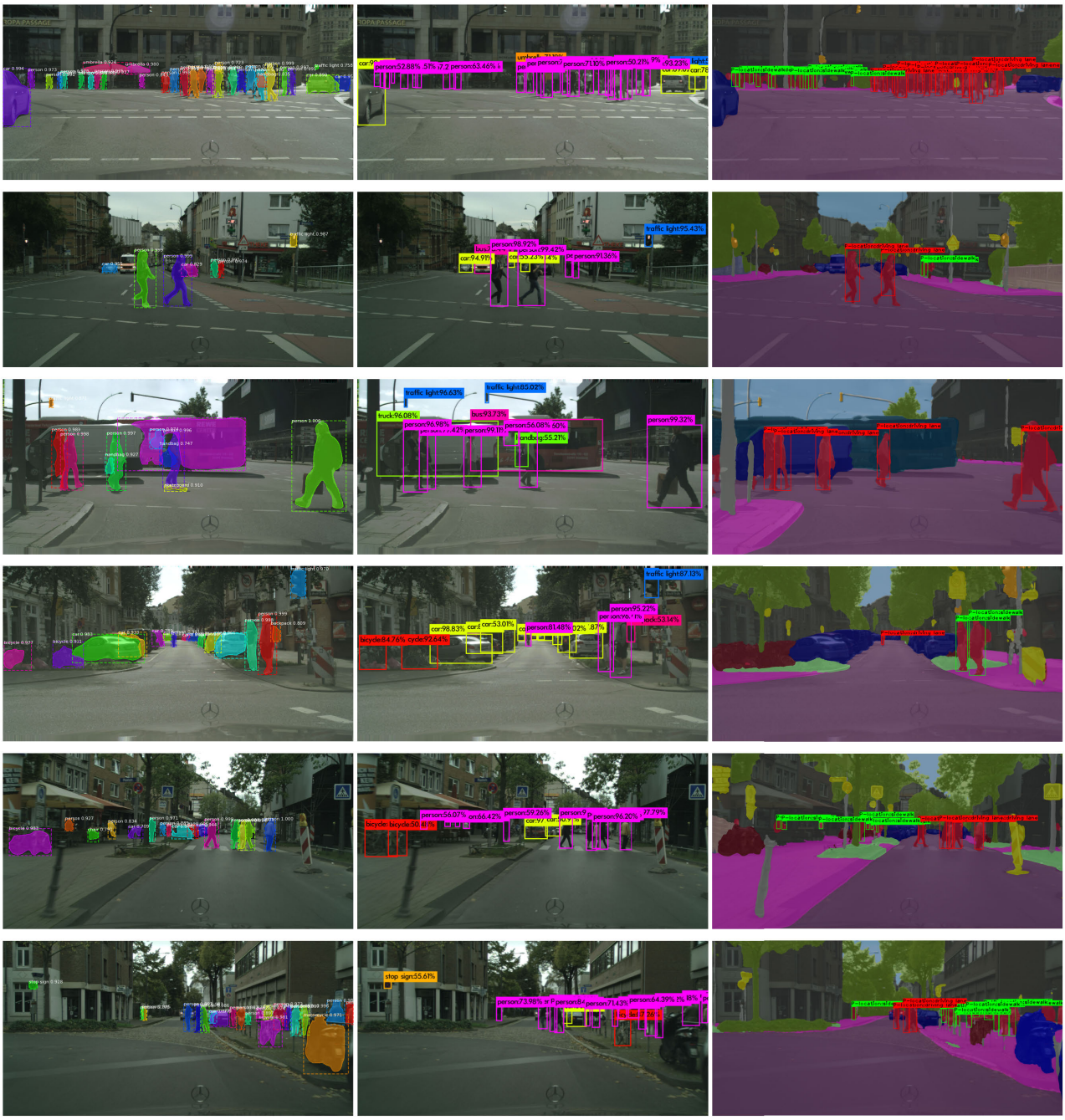


FIGURE 7. Comparison of P-LPN with Mask-RCNN and Yolo V3).

TABLE 1. Comparative results with prevalent methods.

Model	AP	Recall	Time(ms)	fps	mIOU(Segmentation)	mloU(Bounding box)	LRA
PSPNet[14]	n/a	n/a	992	1	**0.80**	n/a	n/a
ICNet [22]	n/a	n/a	**35**	**28.6**	0.69	n/a	n/a
Yolo V3[25]	0.916	0.812	53.2	18.8	n/a	0.75	n/a
Mask-RCNN(Tensorflow)[24]	0.925	**0.838**	4620	0.216	n/a	**0.82**	n/a
P-LPN^{psp}	0.939	0.823	1008	0.99	0.79	0.68	0.925
P-LPN^{inc}	**0.941**	0.826	45 (2nd)	22.2 (2nd)	0.68	0.61	**0.932**

proposed P-LPN has significant advantage for autonomous driving applications.

V. CONCLUSION

In this work, we proposed a high-efficient inner cascade network InCNet for semantic segmentation. Based on this, a novel framework termed "P-LPN" for real-time pedestrian location perception is proposed. Our method enables an end-to-end framework producing semantic segmentation and pedestrian location inference simultaneously. Experiments on CityScapes benchmark demonstrated P-LPN's competitive performance in both accuracy and efficiency. Especially, through customized optimizations, P-LPN can yield real-time inference on complex driving scenes. We believe that P-LPN can be extended to a variety of different autonomous driving applications where fast scene parsing for surrounding objects perception are needed. For further improvement, more work is necessary on integrating pedestrian intention estimation and path prediction methods into P-LPN to enhance its practicality.

REFERENCES

[1] S. Karush. (2018). Status Report: On Foot at Risk. Insurance Institute for Highway Safety Highway Loss Data Institute. Accessed: Oct. 25, 2019. [Online]. Available: https://www.iihs.org/api/datastoredocument/status-report/pdf/53/3

[2] S. Juan. (2016). WHO: 260,000 Die in China as a Result of Road Accidents. Chinadaily. Accessed: Oct. 25, 2019. [Online]. Available: http://www.chinadaily.com.cn/china/2016-05/24/content_25442984.htm

[3] A. Saez, L. M. Bergasa, E. Romeral, E. Lopez, R. Barea, and R. Sanz, "CNN-based fisheye image real-time semantic segmentation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1039–1044.

[4] H. Zhang, A. Geiger, and R. Urtasun, "Understanding high-level semantics by modeling traffic patterns," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3056–3063.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[6] M. Treml, J. Arjona-Medina, T. Unterthiner, R. Durgesh, F. Friedmann, P. Schuberth, A. Mayr, M. Heusel, M. Hofmarcher, M. Widrich, U. Bodenhofer, B. Nessler, and S. Hochreiter, "Speeding up semantic segmentation for autonomous driving," in *Proc. NIPS Workshop (MLITS)*, 2016, p. 7.

[7] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 15, 2019, doi: 10.1109/TITS.2019.2909066.

[8] Z. Ning, Y. Li, P. Dong, X. Wang, M. S. Obaidat, X. Hu, L. Guo, Y. Guo, J. Huang, and B. Hu, "When deep reinforcement learning meets 5G-enabled vehicular networks: A distributed offloading framework for traffic big data," *IEEE Trans Ind. Informat.*, vol. 16, no. 2, pp. 1352–1361, Feb. 2020.

[9] Z. Ning, P. Dong, X. Wang, J. J. P. C. Rodrigues, and F. Xia, "Deep reinforcement learning for vehicular edge computing: An intelligent offloading system," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, 2019, Art. no. 60.

[10] M. Al-Qizwini, I. Barjasteh, H. Al-Qassab, and H. Radha, "Deep learning algorithm for autonomous driving using GoogLeNet," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 89–96.

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

[12] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*. [Online]. Available: http://arxiv.org/abs/1603.04467

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.

[14] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A MATLAB-like environment for machine learning," in *Proc. BigLearn, NIPS Workshop*, 2011, pp. 1–6.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *Comput. Sci.*, vol. 4, pp. 357–361, Dec. 2014.

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[19] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "ERFNet: Efficient residual factorized ConvNet for real-time semantic segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 263–272, Jan. 2018.

[20] C. G. Keller and D. M. Gavrila, "Will the pedestrian cross? A study on pedestrian path prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 494–506, Apr. 2014.

[21] Z. Fang and A. M. Lopez, "Is the pedestrian going to cross? Answering by 2D pose estimation," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1271–1276.

[22] X. Zhang, Z. Chen, Q. M. J. Wu, L. Cai, D. Lu, and X. Li, "Fast semantic segmentation for scene perception," *IEEE Trans Ind. Informat.*, vol. 15, no. 2, pp. 1183–1192, Feb. 2019.

[23] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, M. Jagersand, and H. Zhang, "A comparative study of real-time semantic segmentation for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 587–597.

[24] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "ICNet for real-time semantic segmentation on high-resolution images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 405–420.

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

[26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

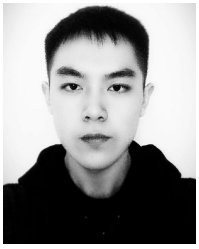
[27] S. W. Keckler, W. J. Dally, B. Khailany, M. Garland, and D. Glasco, "GPUs and the future of parallel computing," *IEEE Micro*, vol. 31, no. 5, pp. 7–17, Oct. 2011.

[28] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

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

[30] 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.

YI ZHAO (Member, IEEE) received the M.Eng. degree from Pierre and Marie Curie University (University Paris VI), France, in 2009, and the Ph.D. degree from the University of Toulon, France, in 2013. He has been with the School of Electronics and Control Engineering, Chang'an University, since 2014. He has also been the Co-Founder and a Chief Scientist with JYI Intelligent-Tech Inc., Shaoxing, since 2018. His current researches mainly focus on machine learning, computer vision, and sensor data fusion. He is currently an Active Reviewer of four IEEE Journals.



MINGYUAN QI received the B.Eng. degree from Chang'an University, China, in 2019, where he is currently pursuing the degree with the School of Information Engineering. His research interests include vehicle network and machine vision in driving scene understanding.

XIAOHUI LI was born in Xi'an, Shaanxi, China, in 1982. He received the B.Eng. degree from the Xi'an University of Technology, China, in 2004, the M.Eng. degree from Paul Sabatier University, France, in 2008, and the Ph.D. degree from the University of Technology of Troyes, France, in 2011.

He has been with the School of Electronics and Control Engineering, Chang'an University, as a Lecturer, since 2012. His research interests include optimization algorithm, scheduling problem, vehicle routing problem, UAV path planning, and multiobjective optimization.

YUN MENG (Member, IEEE) received the B.S. and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 2009 and 2015, respectively. She has been with the School of Electronics and Control Engineering, Chang'an University, since 2015. Her main research interests include interference mitigation and radio resource management in heterogeneous networks and graph theory for wireless networks.

YAXIN YU (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of New Mexico, Albuquerque, NM, USA. Upon graduation, he joined the Singapore's A*STAR Institute of High Performance Computing (IHPC), as a Research Scientist, until June 2017. He is currently an Associate Professor with the College of Electrical and Control Engineering, Chang'an University, Xi'an, China, since July 2017. His research interests include machine learning and neural networks, computational electro-magnetics theory and applications, electromagnetic propagation and phenomena in earth-ionosphere systems, and simulation and modeling of photonic materials and devices. He has served as the Special Session/Symposium Organizer, the Session Chair, and a Technical Program Committee Member for many international conferences. He is also an Active Reviewer of more than 20 technical journals and conferences.

YUAN DONG was born in Xi'an, Shaanxi, China, in 1987. She received the B.E. degree in electronic and information engineering from Northwestern Polytechnical University, in 2009, and the master's and Ph.D. degrees in computer science from the University of Technology of Troyes, France, in 2013. Since 2014, she has been doing her research and teaching with Chang'an University. She has authored more than ten technical articles in refereed journals and conference proceedings. Her main research interests include machine learning, pattern recognition, and wireless hetergeneous networks, especially in study of image processing and decision systems.