IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 6, 2020, accepted September 11, 2020, date of publication September 18, 2020,
date of current version September 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024583

Focus First: Coarse-to-Fine Traffic Sign Detection With Stepwise Learning

LIMAN LIU[1], YUNTAO WANG[1], KUNQIAN LI[2], (Member, IEEE), AND JIE LI[3]
[1]School of Biomedical Engineering, South-Central University for Nationalities, Wuhan 430074, China
[2]College of Engineering, Ocean University of China, Qingdao 266100, China
[3]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China

Corresponding author: Kunqian Li (likunqian@ouc.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976227 and Grant 61906177, in part by the Fundamental Research Funds for the Central Universities under Grant 201813022, in part by the Natural Science Foundation of Shandong Province under Grant ZR2019BF034, in part by the Natural Science Foundation of Hubei Province under Grant 2019CFB622, in part by the Fundamental Research Funds for the Central Universities in South-Central University for Nationalities under Grant CZY17011, and in part by the Key Lab Foundation under Grant 6142113180202.

ABSTRACT Traffic sign detection is a very important part of intelligent assisted driving system. However, with the interference of various target sizes, geometric distortion, occlusion and motion blur, fast and accurate detection on large-size car camera image is extremely hard. To achieve both high efficient and accurate detection, we present a traffic sign detection method within a coarse-to-fine framework, which sequentially detects the targets in grid-level and image-level. We demonstrate that focusing first is a more effective detection strategy for small targets in wide detection space. We propose a target grid prediction network, which is a fully convolutional network for binary classification, to realize rapid coarse localization of the target and effectively guide the clipping and scaling of the target area. With the flexible potential target region extracting strategy, the detecting space can be significantly reduced. At the same time, the correctly extracted local areas for the targets can further facilitate the accurate detection of the subsequent traffic sign detector. In the experiments, our method achieves impressive performance in terms of both efficiency and accuracy. On the challenging Tsinghua-Tencent 100K (TT100K) dataset, our method achieves 20.9 FPS detection speed for 1600 $\times$ 1600 images and the F_1-score of the proposed method for all-scale targets is 91.55.

INDEX TERMS Traffic sign detection, Single Shot MultiBox Detector, convolutional neural network.

I. INTRODUCTION

Traffic signs on the roads convey guidance, warnings, restrictions or instructions with words or symbols, which play important roles in regulating the driving behaviour of drivers and ensuring the safety and smoothness of road traffic. With the improvement of living standards, the public has put forward higher requirements for the safety and convenience of vehicles. So the intelligence of vehicles has been widely concerned. And accordingly, the intelligent detection need for autonomous vehicles is ever growing [1], [21], [35], [51], [59]. A complete intelligent traffic decision-making system needs to perceive the information of traffic signs on the road to make correct decisions. Therefore, effective detection of the road traffic signs is of great importance [30]. However, the detection technology of the traffic signs needs to overcome a series of problems such as geometric distortion caused by the shooting angle, occlusion, deformation of traffic signs and motion blur brought by the high-speed movement of the vehicles. Besides, the diversity and uncertainty of the shape and color of traffic signs make them difficult to be detected. Considering the real-time requirements of intelligent driving and the limited computing ability of on-board unit in vehicles, high efficiency and high precision are equally important for the traffic sign detection algorithms.

Recently, remarkable improvement has been made with the Deep Convolutional Neural Network (DCNN). Successful object detection frameworks, such as Faster-RCNN [39], You Only Look Once (YOLO) [36] and Single Shot MultiBox Detector (SSD) [31], [32], have greatly promoted their practical application. However, directly adopting such general object detection frameworks for traffic sign detection

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

task usually leads to poor performance. For a two-stage framework, such as Faster R-CNN, it is very inefficient to search those small traffic sign targets in the high resolution images. While, SSD is a classical one-stage object detection framework, which has a relatively higher detection speed compared with the two-stage detection approaches but performs poor on small targets. Obviously, these two frameworks can not be directly applied to the traffic sign detection task since it is mainly a small target detection problem with high efficiency requirement. Therefore, a more efficient detection framework with high detection accuracy for small traffic sign targets is needed.

In this paper, a coarse-to-fine traffic sign detection method based on stepwise learning is proposed. It tries to first focus on the coarse locations of the targets and then detect them within their local areas. Our contribution can be summarized as follows:

1) We construct a new two-stage coarse-to-fine traffic sign detection framework, which is demonstrated to be able to effectively improve the detection accuracy and speed.
2) A high efficient potential foreground filtration strategy based on a target grid prediction network is proposed, which can significantly narrow the detection space and receive great efficiency improvement.
3) We design flexible extracting strategies for the potential target regions to effectively cover the whole targets, which facilitate the following detector to receive better detecting performance.

In our detection pipeline, the input image will be firstly divided into grids. Secondly, a lightweight fully convolutional network is designed to give rapid binary prediction for the grids to indicate whether they cover the potential traffic sign targets. The potential target region can be obtained quickly by combining the connected candidate grids according to the criteria of target adhesion. Then, a flexible regional expansion strategy is designed for the target regions with different scales and aspect ratios. With such strategy, we can effectively get proper target contextual regions for the traffic sign detection network. The small target region can be effectively clipped out from the original image and a large target can be fully covered by employing a relative larger contextual region, which can be effectively compressed to a proper target size when we squeeze the region to a 128×128 image for the subsequent detection network. Since the predicted potential location area has been greatly reduced compared with the original image size, a simplified SSD detection network is utilized to quickly identify the target area and deliver the final target detection results. Experiments conducted on the TT100K dataset [59] and the comparison with a series of the latest advanced methods show the good performance of the proposed method.

The rest of the paper is organized as follows. We give a brief review about object detection and traffic sign detection in Section II. The proposed method is introduced in Section III. We present the implementation details and report the comparison results with the state-of-the-art approaches in Section IV. Finally, we conclude our paper in Section VI and give further research prospects.

II. RELATED WORK

A. OBJECT DETECTION

Object detection is the fundamental task in computer vision field, which is also closely related with many other vision tasks, such as object segmentation [25], [26], [45], [46], [50] and object tracking [4], [5], [13], [52]. For decades, object detection has receives tremendous development [60]. As Zou *et al.* mentioned in [60], if we take the successful application of Convolutional Neural Network (CNN) for object detection task as a milestone, object detection can be roughly divided into ''traditional object detection period (before 2014)'' and ''deep learning based detection period (after 2014)''.

In the traditional object detection period, researchers mainly focus on designing sophisticated feature representations and efficient computing strategies. A group of successful hand-crafted features, such as Histogram of Oriented Gradients (HOG) [9], Haar-like features [47] and Deformable Part-based Model (DPM) [12], are proposed and widely used [55], [56].

Due to the limited descriptive abilities of the hand-crafted features, the performance of traditional detectors became saturated quickly. With the fast improvement of computing hardware and the emergence of large scale dataset [22], convolutional neural network receives new development opportunity. After Regions with CNN features (RCNN) for object detection being proposed by Girshick *et al.* [15], the object detection society entered a new era with the wide use of deep convolutional neural network. Since then, numerous excellent detection frameworks are proposed, which can be classified into two categories according to whether they need to produce intermediate detection candidates (object proposals), i.e., two-stage methods and one-stage methods. RCNN series, i.e. RCNN [15], Fast-RCNN [14] and Faster-RCNN [39], Spatial Pyramid Pooling Networks (SPP-Net) [16] and Feature Pyramid Networks (FPN) [27] are typical two-state approaches, while YOLO series (YOLOv1 [36], YOLOv2 [37] and YOLOv3 [38]), SSD [31], [32] and RetinaNet [28] are typical one-stage frameworks. The above methods are general detection frameworks, while traffic sign detection is a more specific task, which involves more practical concerns. In the next subsection, we will give more detailed review for the traffic sign detection task.

B. TRAFFIC SIGN DETECTION

Traffic sign detection algorithms have been studied since the 1990s [30]. At first, traffic signs were detected mainly through the methods based on color and shape. De la Escalera *et al.* [10] proposed a classic RGB threshold segmentation method to detect traffic signs. With the significant success on other computer vision tasks, machine learning

strategies are also widely used in traffic sign detection problem. Before 2012, AdaBoost [48] or SVM [8] based detection methods received impressive success [30]. For example, Zaklouta *et al.* [53] trained SVM to detect traffic targets by utilizing extracted HOG features [9] and achieved a good detection effect. Baró *et al.* designed a boosted detectors cascade for traffic sign detection [3]. However, such hand-crafted features could not further promote the detection performance.

In recent years, deep learning [23] has made rapid development [2], [17], [18], [22], which provides a new strategy to learn suitable features for specified tasks. Accordingly, with the help of task-oriented deep features, object detection algorithm also received continuous breakthroughs in performance [19], [60]. As an important research branch and application field of object detection, traffic sign detection is also deeply affected. Sermanet *et al.* [42] used convolutional networks to classify traffic signs on the German traffic sign detection benchmark and achieved 98.97% classification accuracy. This classification accuracy exceeded the classification ability of human beings. Cireşan *et al.* [6] proposed to combine CNN with hand-crafted features to identify traffic signs and achieved 99.15% identification accuracy on the GTSRB dataset [44]. After that, Cireşan *et al.* [7] constructed a multi-column cascading convolutional neural network, which further refreshed the recognition effect of traffic signs. Jin *et al.* [20] further improved the accuracy of traffic sign recognition by proposing a hinge loss stochastic gradient descent (HLSGD) method to train convolutional neural networks. These algorithms can also achieve good detection accuracy and satisfactory speed on common datasets, such as PASCAL VOC [11], MSCOCO [29] and ILSVRC [40] datasets. However, when it comes to TT100K dataset [59], where vehicle-mounted camera is usually far away from the traffic signs and target usually only occupies a very small proportion of the image, slow detection speed and low detection accuracy will become the main problems.

In TT100K dataset, the portions of small targets (area $\leq 32^2$) and medium targets ($32^2 \leq$ area $\leq 96^2$) are about 40% and 50% of all the targets. Therefore, the above-mentioned tough cases turns the traffic sign detection into a small target detection problem. Recently, several approaches have been proposed to solve small target detection problem. Dealing with small targets, multi-resolution processing strategy is an intuitive and effective strategy. Zhu *et al.* [59] adopted a 8-layer convolutional neural network based on Overfeat framework [41]. It needs to detect the targets on multi-resolution images to compensate the problem of poor scale adaptability. Obviously, this strategy will reduce the detection speed of the algorithm. Liu *et al.* [33] proposed a multi-scale region-based convolutional neural network (MR-CNN) for small traffic sign detection. Additionally, multi-scale contextual regions are extracted in MR-CNN for better target conception. Wang *et al.* [49] presented a small targets detection pipeline within a cascade mask generation framework. The background of multi-resolution images from low to high is removed in a cascading fashion, and then the target in the remaining foreground region will be detected with Faster-RCNN [39]. It not only improves the detection speed of the network, but also improves the detection accuracy. The algorithm reduces the detection range by removing the background in a cascade way and speeds up the detection on the premise of ensuring the target recall. But the real target area is repeatedly detected to adapt to the targets in different scales, which makes the detection process of the algorithm redundant.

Besides, improving small target feature for better detection performance is another interesting strategy. Noh *et al.* [34] proposed to enhance the features of small targets using super-resolution technique, which is demonstrated to be able to improve the detection performance. While, the introduced extra computation cost on feature super-resolution is obvious for the whole detection pipeline. Li *et al.* [24] proposed Perceptual Generative Adversarial Networks (PGAN) to lift the representations of the small targets to the similar characteristics as large objects. This proposal leads to easy location regression and category judgement for the neural network. Although the performance of PGAN has been obviously improved, it is still difficult to achieve high detection speed when apply it to high-resolution images directly.

As adopted in [49], the cascaded model is another popular algorithm structure for fast small targets detection. Zhu *et al.* [58] proposed a detection framework with two deep learning components. One is fully convolutional network based traffic sign proposal generating module and the other is a deep convolutional neural network for object classification. Then, they extended their work for text-based traffic sign detection problem and received good performance [57]. However, computational cost on pixel-wise proposal generation is relatively high.

It can be noticed that, the detection efficiency in traffic sign detection task especially small target detection still needs improving. To cover this concern, we propose a coarse-to-fine detection pipeline, whose front module is a grid-level target prediction network. It significantly promotes the efficiency of coarse location procedure. Besides, to facilitate the following detector to receive better detecting performance, we designed flexible extracting strategies for the potential target regions to effectively cover the whole targets. In the next section, we will give more details of the proposed method.

III. THE PROPOSED METHOD

In this paper, a coarse-to-fine traffic sign detection algorithm based on step-by-step learning is proposed to improve the speed and accuracy of traffic sign detection on high resolution images. The overall flowchart of traffic sign detection algorithm is presented in Figure 1. In the coarse detection stage, the original high-resolution image is first gridded. And secondly, the target grid prediction model with convolutional neural network is used to judge whether a grid contains a traffic sign. Then, in the fine detection stage, the candidate target area is extracted from the positive grids according to the detection results. Finally, the traffic signs will be

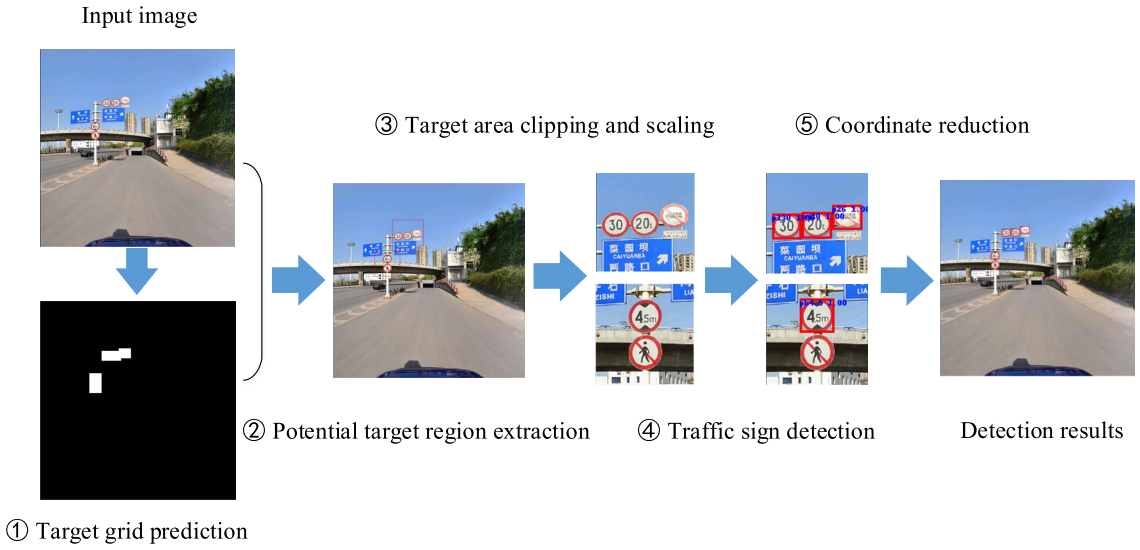
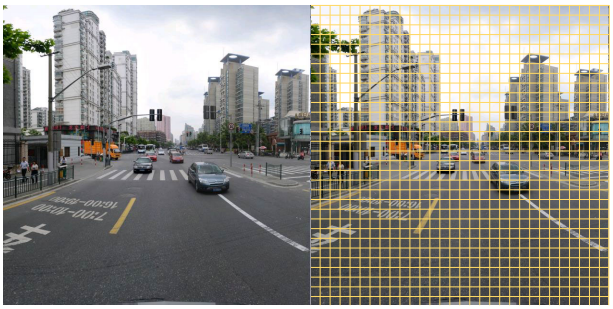


FIGURE 1. The flowchart of the coarse-to-fine traffic sign detection algorithm with stepwise learning.



(a) (b)

FIGURE 2. Example of meshing. (a) raw image. (b) the gridlines represent grid division.

further detected in the potential target area. In the following subsections, we will describe each module with more details.

A. TARGET GRID PREDICTION WITH CONVOLUTIONAL NEURAL NETWORK

The traffic sign usually only occupies a very small proportion of the image, and it will take a long time to directly detect the high resolution image using the conventional target detection algorithm. The large background regions bring most of the futile detection attempts. If the approximate target areas have been extracted beforehand from the large image space, the detecting space of traffic signs will be decreased to those candidate target areas. Then, the efficiency of the traffic sign detection algorithm can be significantly improved. Therefore, in this subsection, we propose a target grid prediction algorithm with convolutional neural network. The grids from the meshed input image will be predicted whether contain traffic signs or not. Accordingly, it will determine the approximate locations of the traffic sign targets through the locations of the positive grids.

1) IMAGE GRID DIVISION

The input image is directly divided into $S \times S$ grids evenly (see Figure 2). Each grid represents a small area in the image, and there is no mutual intersection between different areas, which facilitates the extraction of the target area by the subsequent algorithm. The size of S indicates the fine granularity of grid division, which is related to the target scale range. If the target generally occupies a relatively low proportion of the image, S can be set to a larger value, and vice versa. In practical application, the lower bound of the target size should be more than half of the mesh size to make the mesh contain more target information. Combining these target information with textual information around the grid, the subsequent algorithm can predict the candidate grid with higher accuracy.

2) TRAINING SAMPLE COLLECTION

To collect enough samples for the training of the grid prediction network, we assign the ground truth labels for the grids with the following rules: (a) if the proportion of the real target covering a grid exceeds the threshold t_{ol}, the label of the grid is set to foreground; (b) if the center of the real target is in a grid, its label is also set to foreground; (c) if a grid does not satisfy any of the above positive rules, its label will be set to background.

The above rules are set with the main considerations as follows. Rule (a) is designed for the large-size targets, which tend to occupy one or more grids. Rule (b) is applied to the small-size targets, which can not occupy the setting minimum proportion of any gird. In this case, no grid can match such small target according to Rule (a). Therefore, a grid covering the real target center will be set as a foreground one.

3) DATA AUGMENTATION

To make our training model more robust for complicated real application, we further apply data augmentation,

Algorithm 1 Image Preprocessing for Augmentation

Input : Training image set $\mathcal{I} = \{I_i\}, i = 1, 2, \ldots, N$
Output: Augmented training image set $\tilde{\mathcal{I}} = \{I_i\}, i = 1, 2, \ldots, \tilde{N}$

Initialization: $\tilde{\mathcal{I}} = \mathcal{I}$
for $I_i \in \mathcal{I}$ **do**
  Generate a random probability p_0 which follows uniform distribution in [0,1].
  if $p_0 > 0.5$ **then**
    Generate a random probability p_1 which follows uniform distribution in [0,1].
    if $p_1 > 0.5$ **then**
      Do random brightness adjustment: $\tilde{I}_i^b = \mathcal{B}(I_i)$.
      Add $\tilde{I}_i^b$ into $\tilde{\mathcal{I}}$.
    end if
    Generate a random probability p_2 which follows uniform distribution in [0,1].
    if $p_2 > 0.5$ **then**
      Do random saturation adjustment: $\tilde{I}_i^s = \mathcal{S}(I_i)$.
      Add $\tilde{I}_i^s$ into $\tilde{\mathcal{I}}$.
    end if
    Generate a random probability p_3 which follows uniform distribution in [0,1].
    if $p_3 > 0.5$ **then**
      Do random contrast adjustment: $\tilde{I}_i^c = \mathcal{C}(I_i)$.
      Add $\tilde{I}_i^c$ into $\tilde{\mathcal{I}}$.
    end if
  end if
end for

i.e., introducing random variation to simulate different illumination conditions and target sizes.

We adjust the brightness, saturation and contrast of the image to adapt to different light and weather. Specifically, the algorithm keeps the original image with 50% probability and adjusts it with 50% probability; if adjustments are required, the brightness, saturation and contrast are adjusted randomly within the given variation range with a 50% chance (see Algorithm 1). In our implementation, we adopt the default random adjusting parameters of the original SSD. The algorithm uses the IoU based sampling strategy of SSD to construct multi-scale training samples by randomly adjusting the size of the sampled image blocks, so as to improve the recognition accuracy of the network at each scale.

4) HARD NEGATIVE MINING

Considering that the traffic signs only occupy very small parts of the input image, it will introduce significant imbalance between the positive and negative training examples. Such unbalanced training set will greatly reduce the generalization of deep model or even lead to the failure of model training. Therefore, in our implementation, all the positive samples are included for the training, while only an equal number of error-prone negative samples are selected to establish the training set.

TABLE 1. Structure of grid prediction network.

VGG-T	VGG-S
conv3-64 conv3-64	conv3-32 conv3-64
maxpooling	
conv3-128 conv3-128	conv3-128
maxpooling	
conv3-256 conv3-256 conv3-256	conv3-256
maxpooling	
conv3-512 conv3-512 conv3-512	conv3-512
normalized layer	normalized layer
conv3-2	conv3-2

5) TARGET GRID PREDICTION

To realize the foreground prediction of the grid, a convolutional neural network based prediction model is constructed. The approximate position of the target is determined by the position of the foreground grid in the prediction result.

The grid prediction network is a simple fully convolutional network for binary classification, whose structure is presented in Table 1. The backbone network adopts the truncated model (VGG-T) of VGG-16 [43], that is, only the first four stages of VGG-16 are used for grid prediction. We remove the activation layer of conv4_3 and set a normalized layer instead. Then, the final convolution layer is utilized to give a foreground or background classification for each grid. In addition, as presented in right column of the Table 1, we further simplify the above truncated model to speed up the prediction, which is called the simplified model of VGG-16 (VGG-S).

6) LOSS FUNCTION

The grid prediction networks employs the softmax loss as follows:

$$L = \frac{1}{N}(L(+) + \alpha L(-)), \tag{1}$$

where the overall loss L is the weighted sum of the positive sample loss $L(+)$ and the negative sample loss $L(-)$, and α is the weighting coefficient to modulate the network's attention to positive and negative samples. N is the number of positive samples in the grid area. If $N = 0$, the total loss is set to 0. The positive and negative sample losses can be computed as:

$$L(+) = -\sum_{i \in \text{pos}}^{N} x_i^+ \log(c_i^1), \tag{2}$$

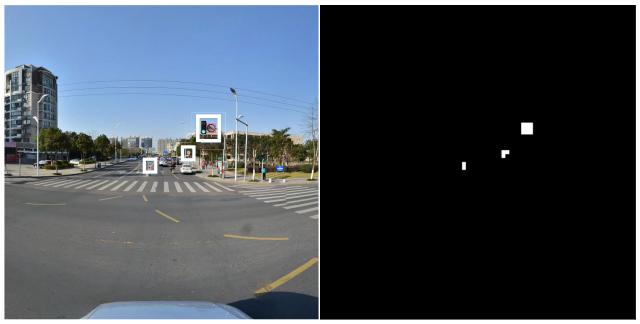


FIGURE 3. Example of grid prediction results. The left is the raw image, in which the white box indicates the approximate location of the target. On the right side, the prediction results of the grids are shown. The white in the right image indicates that the corresponding grid is predicted to be the foreground, and the black indicates that the corresponding grid is predicted to be the background.

$$L(-) = -\sum_{i \in \text{neg}}^{N} x_i^- \log(c_i^0), \tag{3}$$

where $x_i^+ = \{0, 1\}$ denotes whether the i-th grid area is a positive sample. If it is a positive sample, $x_i^+ = 1$; otherwise, $x_i^+ = 0$. $x_i^- = \{0, 1\}$ denotes whether the i-th grid area is a negative sample participating in the training. If so, $x_i^- = 1$; otherwise, $x_i^- = 0$; c_i^p denotes the probability that the i-th grid area is on class p.

B. POTENTIAL TARGET REGION EXTRACTION

Example results of grid prediction are shown in Figure 3. It can be noticed that the connected positive grid regions can roughly reflect the position of the real targets in the raw image.

1) CONNECTED POSITIVE GRIDS EXTRACTION

The above grid prediction model classifies all the grids into foreground and background ones, but the prediction results are still discrete labels. By analysing the grid label designation rules, we can find that a real traffic sign target may cover one or more foreground grids. And at the meantime, the predicted positive grids belonging to the same target must be connected on the 8-neighborhood. Therefore, the location of one or more targets can be determined roughly by merging adjacent positive grids and extracting the connected regions of the positive grid in the predicted image. In this paper, we adopt a depth-first searching strategy to determine the adjacency relation of the positive grids and record all the foreground regions.

2) POTENTIAL TARGET REGION EXPANSION

According to the grid prediction results, the external rectangle of the connected region can be viewed as the potential target region, which can roughly reflect the location of the target in the image. However, the target area is not guaranteed to contain the target completely. It is mainly because the training positive samples need to meet one of the above-mentioned conditions, i.e. large foreground covering or target center lying in the grid. For large target, the edge part of the target may only cover a relatively small proportion of the grid, which may lead to a background prediction. While for a small target, the grid division may divide the target into different grids, which may also lead to different predictions for these grids. Therefore, the potential target region needs to be expanded properly.

We design different inflation strategies for different target regions according to their sizes and aspect ratios. The initial potential target region is an externally connected rectangle area of the extracted connected grid region. If the longer and shorter sides of the rectangle are denoted as w grids and h grids, respectively, the designed expansion strategy can be described as follows:

1) If the target region is small, i.e., the region length $w \leq l_{th}$, then we directly expand the region to $M \times M$ (grids) to keep the detailed characteristics of the target. l_{th} is the region length threshold, which is set to 4 girds in our implementation. M is defined as

$$M = S_{input}/S_{grid}, \tag{4}$$

where S_{input} is the default size of the input image for the subsequent detection module, S_{grid} is the size of each grid in the original image.

2) If the potential target region is a large long strip, i.e., $w > l_{th}$ and $w/h > r_{th}$, we expand the region to $(w + 1) * (w + 1)$. r_{th} is the region length width ratio threshold, which is set to 3 in our implementation. This is mainly designed for the case where a potential target region covers multiple traffic signs. In this case, a single traffic sign is not large compared with the whole area, only a small expansion of the area boundary is required to ensure that the target area can fully contain the target.

3) If the potential target region is a large square block, i.e., $w > l_{th}$ and $w/h \leq r_{th}$, then we expand the region to $(r * w + 1) * (r * w + 1)$, where r is the target scale adjustment parameter and $r = 1.2$ in our implementation. This setting focuses on the single large target. The boundary is first expanded by a small range to ensure that the region contains the target completely. Then, we proportionally expand this potential target region to shrink the visual size of the large target in the potential region.

Besides, after the expansion of the target regions with the above rules, the target region may exceed the boundary. For example, when the traffic sign is at the image boundary, the region needs to move towards the inside of the image. Specifically, if the region is out of bounds on the left, the region will be moved to the right until the left boundary of the region coincides with the left boundary of the image. Examples of region expansion in the TT100K dataset is presented in Figure 4.

All the above-mentioned sizes in region expansion are all grid-based. To further clip such regions out of the original image for detection, we should first project such grids to

FIGURE 4. Examples of the potential target region expansion. The blue box is the original potential target area, and the red box is the region after expansion. The left picture is an example of inflating large target regions; the middle is an example of inflating small foreground regions; and the right is an example of shifting an out-of-bound box.

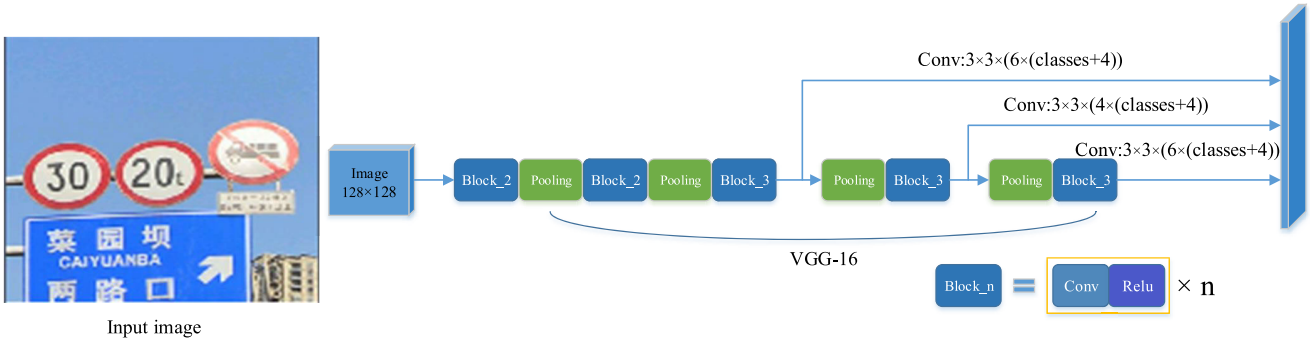


FIGURE 5. Detection network. The activation layers after Conv3_3, Conv4_ 3 and Conv5_3 layers are removed. Each of them is connected with a normalization layer and two convolutional layers to predict the class and location of the target, respectively. Block_n means there are *n* "Convolutional layer – Relu layer" modules in the block.

the original image and get the corresponding region. Then, we resize them the required input size of detection network, which is 128 × 128 in our method (see Section III-C).

C. TRAFFIC SIGN DETECTION

With the obtained potential target regions, we further detect the traffic signs within a simplified SSD object detection framework [32].

1) NETWORK STRUCTURE AND DEFAULT BOXES SETTING

The traffic sign detection framework adopts the SSD multi-scale network, and the trunk network selects VGG-16 model to detect the targets directly on the output feature layer of Conv3_3, Conv4_3 and Conv5_3 of VGG-16 (see Figure 5). The input image of the detecting network is set to 128 × 128. Setting such a relatively small image size not only ensures the detection speed, but also keeps enough details of both small and large targets. After the image scaling, the approximate range of target sizes is [16, 106]. With consideration of the perceptive field of each feature layer, we set the sizes and shapes of the default boxes. The size of the perceptive field and the setup of the default box settings at each stage are shown in Table 2.

2) HARD NEGATIVE MINING

As with the selection strategy in Section III-A, the proportion of the positive and negative samples for training is set to 1:1. The negative samples with the bad predictions will be selected to participate in the training every time. In this way, when the selected default boxes can be well predicted, the prediction of the overall negative samples will be satisfactory.

3) LOSS FUNCTION

By reference to the design of SSD [32], the total loss of the model is defined as the weighted sum of the localization loss of positive samples and the confidence loss of positive and negative samples:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)), \qquad (5)$$

where N denotes the number of positive samples and the weight term α is set to 4.0 by cross validation. The localization loss is the Smooth $L1$ loss between the predicted box l and the ground truth g. It regresses to the offsets for the center $(cx,\ cy)$ of the default bounding box d and for its width w and

TABLE 2. The receptive field sizes and default box settings at different stages of VGG-16. The aspect ratios of the default box are set to two proportions, i.e., 1.0 and 2.0. It means that only square and vertical rectangle are set, which depends on the camera angle when the car is running. In actual traffic scenes, a traffic sign is generally a regular polygon and cars are often photographed at elevation angles, which makes the target shape may be deformed and become a partial vertical rectangle.

feature layer	Conv1_2	Conv2_2	Conv3_3	Conv4_3	Conv5_3
receptive field	5	14	40	92	196
sizes	–	–	15, 20, 30	48, 64	72, 90, 110
aspect ratios	–	–	1.0, 2.0	1.0, 2.0	1.0, 2.0

height h:

$$L_{\text{loc}}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} smooth_{L1}(l_i^m - \hat{g}_j^m), \quad (6)$$

$$\begin{cases} \hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w, & \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h, \\ \hat{g}_j^w = \log(\frac{g_j^w}{d_i^w}), & \hat{g}_j^h = \log(\frac{g_j^h}{d_i^h}), \end{cases} \quad (7)$$

where $x_{ij}^k = \{0, 1\}$ denotes whether the i-th default box is matched to the j-th real target box of class k. When the two boxes are matched, i.e. their jaccard score is larger than 0.5, $x_{ij}^k = 1$, and $x_{ij}^k = 0$, otherwise. In the localization loss, only the positive samples are considered.

The confidence loss is the softmax loss over multiple classes confidences (c), which is defined as

$$L_{\text{conf}}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg}^{RN} \log(\hat{c}_i^0), \quad (8)$$

where $\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$ denotes the classification confidence of this box over class p. More detailed information can be found in [32] and [31].

IV. EXPERIMENTAL RESULT

A. DATASETS AND EVALUATION CRITERION

The image size of Tsinghua-Tencent 100K(TT100K) dataset is 2048 × 2048, but the sizes of most targets are between 32 pixels and 96 pixels. It means that each traffic sign target only occupies a very small proportion of the whole image. Therefore, we select TT100K as the training and testing dataset to evaluate the algorithm performance on the complicated real scenarios. We select 45 classes whose instance number exceed 100. The targets are classified into small-scale (target area $< 32^2$ pixels), meso-scale (32^2 pixels $<$ target area $< 96^2$ pixels) and large-scale (target area $> 96^2$ pixels). F_1-score is used as the overall evaluation index, which can comprehensively reflect the detection results of the algorithm in recall and accuracy. When F_1-score is higher, the detection effect of this method is better. F_1-score is computed is:

$$F_1\text{-score} = 2 * \frac{precision * recall}{precision + recall}, \quad (9)$$

where *precision* is the accuracy of test results and *recall* is the recall rate of test results.

B. SPECIFIC IMPLEMENTATION DETAILS

Our overall framework is mainly composed of target grid prediction and traffic sign detection. The whole network is trained in two stages. For the target grid prediction network, the training image size is 400 × 400, the test image size is 640 × 640, gird dividing parameter S is set to 64, the overlapping threshold of target and grid t_{ol} is 0.05. VGG-S is used as the backbone of the grid prediction network. The training process was optimized by Stochastic Gradient Descent (SGD) with 0.9 momentum, 0.0005 weight decay and batch size 8. We first train the network with a 10^{-3} learning rate for 200k iterations, then continue the training for 40k iterations with a 10^{-4} learning rate and 20k iterations with a 10^{-5} learning rate. We use the hard negative mining strategy to balance the number of positive and negative samples. The ratio between the negatives and positives after balance is 1 : 1, and the loss coefficient for the positive and negative sample is set to 2 : 1.

For the training of traffic sign detection network, the size of input image is set to 128 × 128, the training process is also optimized using SGD with 0.9 momentum, 0.0005 weight decay, and batch size 6. And the initial learning rate is 10^{-3} with 100k iterations, followed by 20k iterations with 10^{-4} and 10^{-5} learning rate, respectively. The ratio of positive and negative sample is set to 1 : 1, and the ratio of classification loss and localization loss coefficient is set to 4 : 1.

The training process is performed on a single NVIDIA GTX TITAN X GPU.

C. EXPERIMENTAL RESULTS OF THE OVERALL ALGORITHM

We evaluate the performance of the proposed method on the TT100K dataset and compare it with RefineDet [54], the methods of Zhu *et al.* [59], Li *et al.* [24], Wang *et al.* [49], Liu *et al.* [33] and Noh *et al.* [34], which are the latest advanced approaches for traffic sign detection. We follow the protocol of the [59] to evaluate for 45 classes that include more than 100 instances. The overall quantitative evaluation results are presented in Table 3 and the running time comparison is summarized in Table 4.

The comparison results show the advantages of the proposed method on both accuracy and efficiency. Our method achieves a 91.55 score in overall detection accuracy, which is the second best performance among all the seven methods. Particularly, for the detection of small targets, our method outperforms all the comparison methods except [34]. This is because that the proposed target grid prediction module

TABLE 3. Traffic sign detection results at each scale range (F_1-score/rank).

Target size	Small	Medium	Large	All
RefineDet [54]	77.77 / **7**	84.39 / **7**	33.48 / **7**	78.38 / **6**
Zhu et al. [59]	84.39 / **5**	92.69 / **6**	88.99 / **6**	89.17 / **5**
Li et al. [24]	86.43 / **3**	93.43 / **5**	89.99 / **5**	90.43 / **3**
Wang et al. [49]	82.20 / **6**	95.89 / **2**	93.91 / **2**	89.68 / **4**
Liu et al. [33]	86.00 / **4**	93.50 / **4**	90.10 / **4**	- / -
Noh et al. [34]	88.60 / **1**	96.00 / **1**	95.40 / **1**	93.10 / **1**
Ours	86.73 / **2**	94.95 / **3**	92.67 / **3**	91.55 / **2**

TABLE 4. Algorithm running speed comparison.

Algorithm	RefineDet [54]	Zhu et al. [59]	Li et al. [24]	Wang et al. [49]	Ours
Average speed (FPS)	2.5	5	<1.6	9.6	20.9

TABLE 5. Average time cost of each module of the proposed algorithm framework.

Test phase	Target grid prediction (on GPU)	Target region extraction (on CPU)	Cut zoom (on CPU)	Target detection (on GPU)	Non-maximum suppression (on CPU)	Total time
Average time	20.6ms	6.6ms	1.9ms	8.7ms	10.1ms	47.9ms

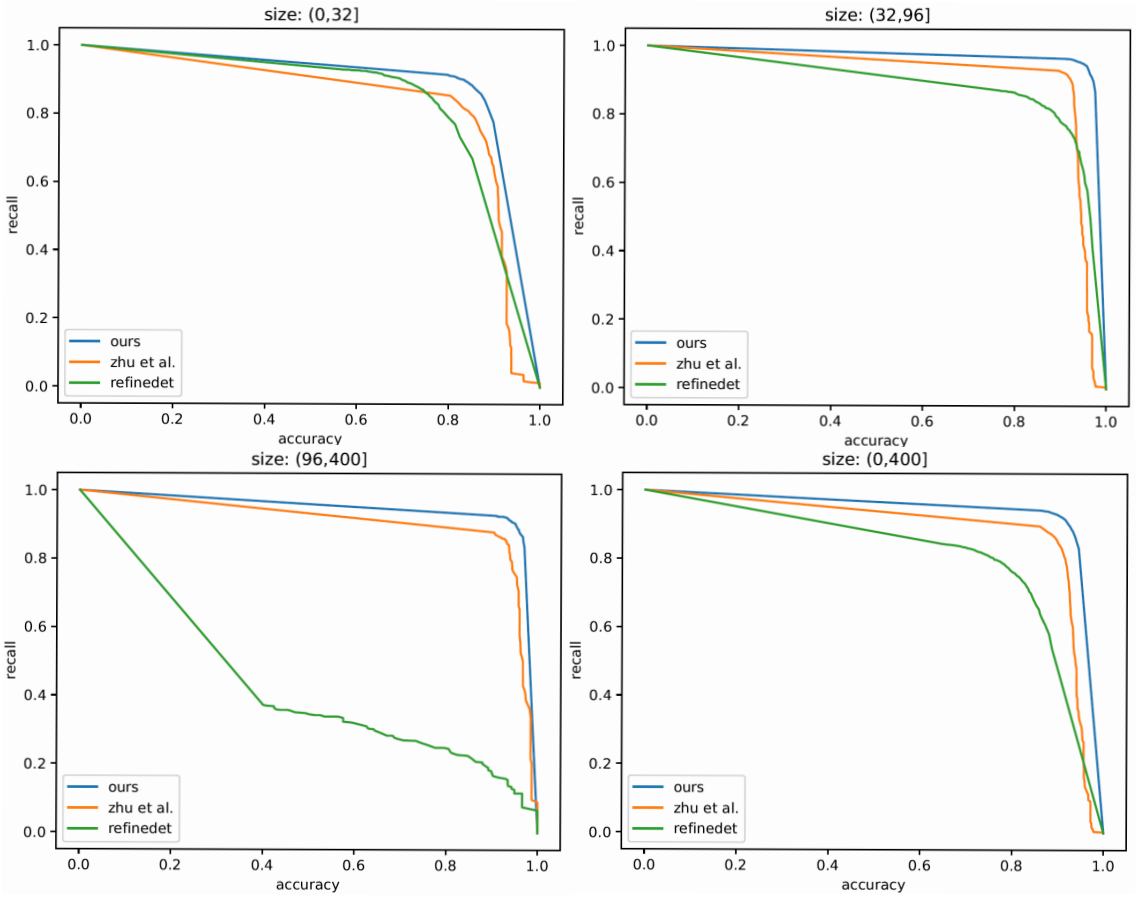


FIGURE 6. Accuracy-recall curves for various size ranges. The curves show the performance changing under different t_p values. Performance curves of RefineDet [54], [59] and the proposed method are shown in green, orange and blue, respectively.

can point out the potential target region effectively, which significantly facilitates the appropriate clipping and scaling of the small target region. Then, those small traffic signs can be effectively detected in their local regions. Due to the significantly improved RoI features with super-resolution technique, [34] achieves the best overall detection performance. For the medium-scale and large-scale targets, compared with approaches of Zhu *et al.* [59], Li *et al.* [24] and Liu *et al.* [33], the performance advantage of our method is also very obvious. It is mainly because that the target

TABLE 6. Comparison of the detection performance for each traffic sign category. The performance of RefineDet [54], methods introduced in [24], [59] and the propose method are presented. The best performance for each category is presented in bold font. (R): Recall, (A): Accuracy. (In %).

Class	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	p19	p23	p26	p27
RefineDet [54] (R)	71.4	85.7	94.5	76.9	68.8	48.4	73.9	89.7	79.1	87.8	29.2	74.2	70.4	72.3	55.3
RefineDet [54] (A)	68.0	84.6	88.0	96.8	91.3	90.2	66.1	70.6	76.4	76.1	63.3	67.6	88.5	73.8	92.9
Zhu et al. [59] (R)	82.4	93.5	95.0	97.4	91.4	93.8	88.7	92.2	95.4	91.1	89.4	93.9	94.2	93.0	95.8
Zhu et al. [59] (A)	72.2	82.8	91.6	100	90.8	92.8	75.9	87.4	78.3	89.2	88.1	88.1	87.4	81.8	78.0
Li et al. [24] (R)	84	95	95	**95**	**92**	**95**	92	**91**	89	96	97	97	95	**94**	**98**
Li et al. [24] (A)	85	92	94	**97**	**95**	**83**	79	**90**	84	85	88	84	92	**83**	**98**
Ours(R)	**85.2**	**96.1**	**94.6**	79.5	77.9	78.1	**89.4**	92.2	**94.3**	**96.0**	**95.5**	**93.9**	**97.1**	84.6	93.6
Ours(A)	**88.9**	**92.5**	**95.0**	96.9	97.3	94.9	**85.0**	88.7	**86.3**	**89.0**	**94.0**	**88.6**	**92.6**	87.6	93.6

Class	p3	p5	p6	pg	ph4	ph4.5	ph5	pl100	pl120	pl20	pl30	pl40	pl5	pl50	pl60
RefineDet [54] (R)	39.7	92.5	7.7	93.2	33.3	54.2	51.3	73.4	63.1	30.9	67.7	84.9	78.5	84.0	69.3
RefineDet [54] (A)	95.8	86.0	50.0	56.9	66.7	71.1	76.9	95.7	94.6	85.0	78.2	84.3	79.7	78.1	84.2
Zhu et al. [59] (R)	91.4	95.1	87.2	90.9	81.6	**88.3**	**82.1**	**98.1**	**97.7**	96.4	94.1	95.5	93.5	94.1	93.4
Zhu et al. [59] (A)	80.3	88.5	87.2	93.0	93.9	**88.3**	**88.9**	**96.8**	**100**	90.0	90.2	89.3	83.6	87.0	92.8
Li et al. [24] (R)	**93**	96	100	**93**	78	88	85	96	98	96	93	96	92	96	91
Li et al. [24] (A)	**92**	90	83	**93**	97	68	69	97	98	92	91	90	86	87	92
Ours(R)	81.0	**99.2**	**89.7**	95.5	**83.8**	86.7	82.1	85.6	86.2	**94.6**	**97.1**	**95.3**	**91.0**	**96.5**	**95.3**
Ours(A)	87.0	**91.6**	**97.2**	84.0	**100**	86.7	86.5	96.9	100	**96.4**	**92.6**	**93.8**	**91.5**	**90.1**	**95.3**

Class	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	wo
RefineDet [54] (R)	34.1	71.9	42.9	46.9	63.2	97.5	97.8	60.6	68.3	45.2	24.2	75.4	80.0	69.0	10.8
RefineDet [54] (A)	100	83.1	87.5	88.2	75.0	75.9	84.9	58.9	83.7	56.0	53.3	58.9	75.6	57.1	25.0
Zhu et al. [59] (R)	**93.1**	**95.3**	87.8	90.6	94.9	91.1	93.2	67.1	98.4	64.5	70.6	71.7	79.0	82.0	44.7
Zhu et al. [59] (A)	**95.3**	**94.3**	91.5	80.6	59.7	92	93.3	84.3	75.6	64.5	88.9	86.0	95.1	74.6	51.5
Li et al. [24] (R)	91	99	88	94	100	**96**	97	83	**97**	94	85	**95**	94	95	53
Li et al. [24] (A)	97	86	90	77	81	**89**	93	78	**92**	66	83	**88**	93	71	54
Ours(R)	88.6	92.4	**93.9**	**90.6**	**94.7**	95.4	**95.6**	**80.5**	96.8	**96.8**	**85.3**	88.3	**95.1**	**95.0**	**57.9**
Ours(A)	97.5	93.0	**100**	**100**	**94.7**	89.3	**95.5**	**81.5**	87.1	**85.7**	**85.3**	85.5	**93.5**	**72.2**	**62.9**

expansion strategy is flexibly utilized for the target areas in different scales to better control the target scale within a reasonable range, which will be easier to be identified by the detection network. Our detection performance for medium-scale and large-scale targets is only behind the method of Wang *et al.* [49] and state-of-the-art [34], but our method is more efficient than its competitors.

According to the algorithm running speed comparison results presented on Table 4. Our approach achieves 20.9 FPS detection speed on NVIDIA TITAN X GPU, while the speed of the other three algorithms except method introduced in [49] are no more than 5 FPS. The detection speed of [49] is reported as 9.6 FPS on NVIDIA GTX 1080Ti GPU, whose computing power is basically the same with NVIDIA TITAN X GPU. The proposed method achieves 2x detection speed compared with [49]. Such high detection speed benefits from the grid prediction network, which can quickly locate the potential target region on the low-resolution image. Different with [49], which needs to repeatedly detect the potential target to adapt to multi-scale targets, the proposed grid prediction process can spend less time filtering out most of the background regions in the image and make the target detection process more directly. Therefore, the overall detection time can be significantly shorten. Above all, among all the comparison methods, our method is the only method which can simultaneously achieve good efficiency and accuracy.

To further analyse the time cost of each module in the whole detection process, we split the algorithm flow and count the running time of each module on the test dataset to facilitate the subsequent further optimization of the detection speed. The algorithm is implemented within TensorFlow framework and written with the Python language. Additionally, the GPU is NVIDIA GeForce GTX TITAN X and the CPU is Intel Xeon E5-2633. It can be noticed from Table 5, the main time consumption lies in the target grid prediction part. In the future, introducing prior knowledge of the spatial position to remove those impossible regions, such as the higher sky and the near ground, could be a promising improvement strategy for faster detection.

To further analyse the detection performance of each algorithm under different classification thresholds t_p, we plot the recall-accuracy values of several methods under different t_p values in Figure 6. Since the projects for the works in [24], [33], [34] and [49] are not publicly available, we only plot curves of RefineDet [54], [59] and the proposed method. It can be noticed that the performance curve of our algorithm is much closer to the upper right of the figure on all scale ranges ($0 \sim 400$). With any classification threshold, the pro-

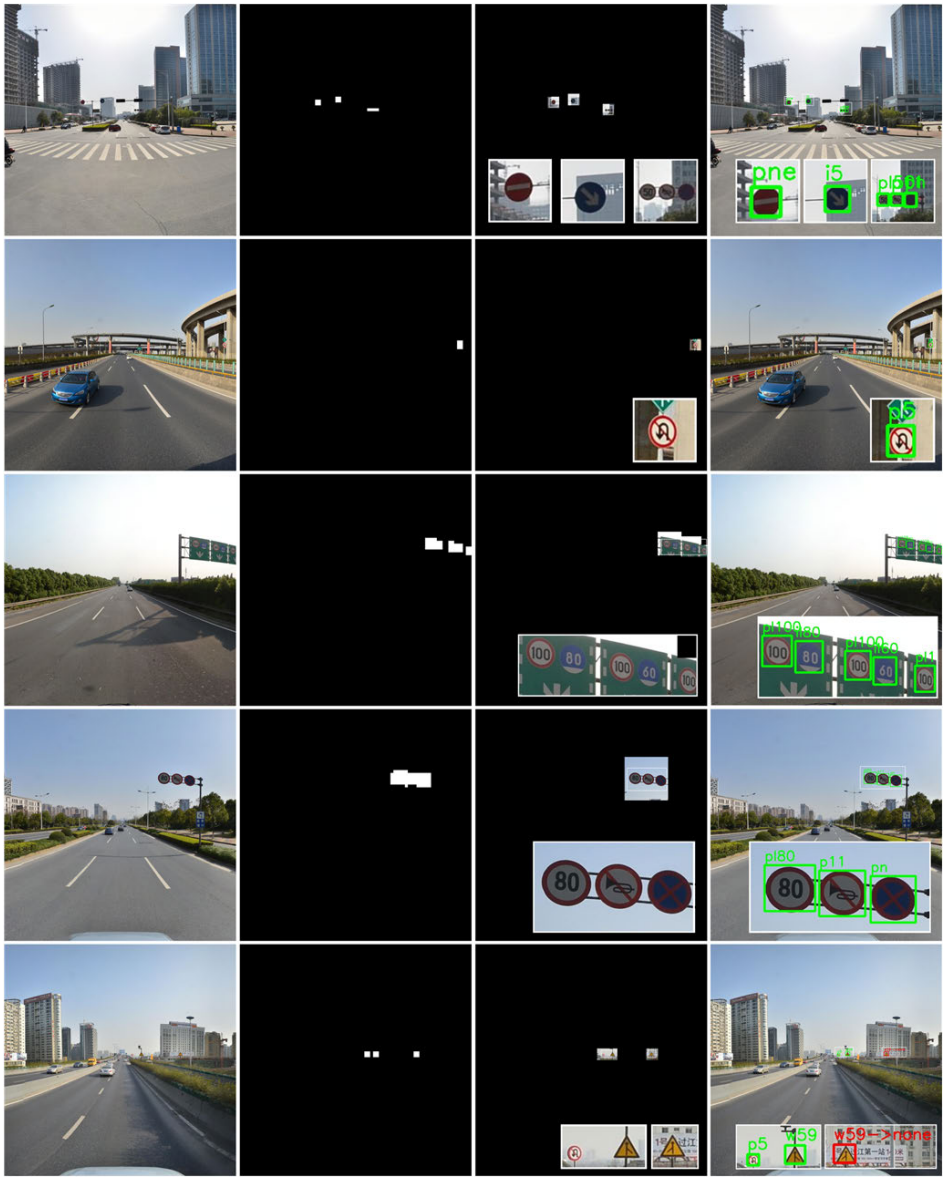


FIGURE 7. The visualization of the algorithm data flow. The first column shows the input original images. The second column presents the results of target grid prediction. The third column presents the extracted potential target regions. We enlarge the region in the lower right corner of the image for clarity. The final detection results are shown in the fourth column. We also enlarge the final detection results in the lower right corner for clear display. In the detection result, the green rectangle boxes are the correctly identified targets (True Positive); the red rectangle boxes are the incorrectly identified rectangle boxes (False Positive).

posed algorithm can produce relatively better detection results in the whole scale range and three subdivided scales.

In Table 6, we also give the recall and accuracy of comparison methods [24], [54], [59] for each kind of traffic sign. For clarity, the items with the dominant-performance in each class are bolded. Considering that it is a two-fold comparison according to both recall and accuracy, the F_1-score is used as the standard to judge the overall performance of the comparison methods for each category. As can be seen from the table, the proposed method performs the best, which can achieve the best detection performance in most categories. Specifically, our method can achieve the best performance on 28 categories of all the 45 categories, while the method of Li *et al.* [24] performs the best on 11 categories and the method of Zhu *et al.* [59] performs the best on only 6 categories.

However, our method has low recalls for the minimum speed traffic signs, such as il100, il60 and il80. This is not surprising because these traffic signs mostly appear in highway scenarios, which are usually far from the cameras and arranged in lines with multiple signs. After forecasting the foreground and background labels for the grids with the target grid prediction module, the target region extraction algorithm may put multiple targets into a single target region. As a result, the target size will be too small to accurately identify after the region being compressed to 128 × 128.

D. VISUALIZATION OF THE ALGORITHM DATA FLOW

In Figure 7, to further clearly present the data flow of the proposed algorithm, we visualize some intermediate results of each stage while detecting the test images. It can be seen from the figure that, the target grid prediction algorithm has a good identification of positive and negative grid samples, which greatly narrows the searching field of the following detection. The extracted target areas can better indicate the approximate position of the real targets and make the targets covered completely. From the final detection results, we can notice that the overall algorithm can get the correct detection for most of the targets.

V. DISCUSSION

We propose a two-stage framework for traffic sign detection in this paper. But, compared the traditional two-state deep learning based detectors, such as Fast-RCNN and Faster-RCNN, our strategy has obvious difference with them.

First, the purpose and subsequent processing of potential target region is different with that of proposals in Fast-RCNN or Faster-RCNN. Proposals generated by Fast-RCNN/Faster-RCNN are expected to indicate the target locations as accurate as possible. Then, deep regression is utilized to refine the proposal locations. While, as presented in Figure 4, our potential target region extraction strategy is to locate the potential target within a proper image context. Then, within the potential target region, more accurate locations can be achieved by sign target detectors with the help of rich contextual information.

Secondly, the generation method of potential target region is totally different with that of proposals in Fast-RCNN/Faster-RCNN and more suitable for small target detection task. In Fast-RCNN, proposals are generated with selective search, which is very time-consuming and insensitive to small target. Although RPN proposed in Faster-RCNN improves the speed, its anchor-based proposal generation strategy is still very inflexible when detect targets with arbitrary sizes, especially small target. In this paper, we propose to train a target grid prediction network to locate the potential target region. It greatly increases the flexibility since the grids, as the minimum region elements, can form target regions with arbitrary sizes according to the prediction results. Moreover, the lightweight grid prediction network is very efficient and guarantees the high detection speed.

Finally, we design flexible potential target region expansion strategy, which can resize the targets to moderate scales by expanding the target contexts before inputting them into detection network. Therefore, the detection accuracy also is improved.

Extremely large and small targets are tough issues in object detection task. While, in a general street scene, compared with the whole image, traffic signs photographed by the in-vehicle cameras would not be extremely large. Fortunately, for a relatively large traffic sign, it can be effectively covered by multiple girds. Besides, in our implementation, we set an appropriate grid size to make the lower bound of the target size to be more than half of the grid size. It can effectively get the small targets to be covered and detected.

VI. CONCLUSION AND FUTURE WORKS

In this paper, a coarse-to-fine traffic sign detection algorithm with stepwise learning strategy is proposed to improve the detection speed and accuracy for high-resolution images and small targets. Firstly, the proposed algorithm meshes the input image, then uses the target grid prediction network to detect whether the grid contain a target. According to the gird prediction results, we flexibly extract the potential target regions to completely cover the real targets. Finally, SSD is utilized to detect the traffic signs on the potential target regions. The target grid prediction is based on a concise convolutional neural network, which can quickly and accurately locate the approximate position of the target and remove the background from the searching space. This strategy get the overall algorithm greatly accelerated. Experiments on the challenging TT100K dataset verify the effectiveness and efficiency of the proposed algorithm. In the future, we will explore the application of spatial prior knowledge in faster foreground region filtering and design specific processing strategies for those tightly arranged small traffic signs.

REFERENCES

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[2] 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.

[3] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary AdaBoost detection and forest-ECOC classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 113–126, Mar. 2009.

[4] K. Chen and W. Tao, "Once for all: A two-flow convolutional neural network for visual tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3377–3386, Dec. 2018.

[5] K. Chen and W. Tao, "Learning linear regression via single-convolutional layer for visual object tracking," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 86–97, Jan. 2019.

[6] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1918–1921.

[7] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

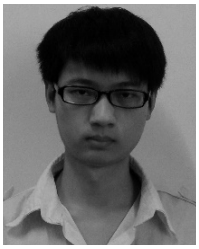
[10] A. de la Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," *IEEE Trans. Ind. Electron.*, vol. 44, no. 6, pp. 848–859, Dec. 1997.

[11] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] Q. Feng, J. Huang, and Z. Yang, "Jointly optimized target detection and tracking using compressive samples," *IEEE Access*, vol. 7, pp. 73675–73684, 2019.
[14] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.
[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[16] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[17] 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.
[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[19] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.
[20] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.
[21] U. Kamal, T. I. Tonmoy, S. Das, and M. K. Hasan, "Automatic traffic sign detection and recognition using SegU-net and a modified tversky loss function with L1-constraint," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1467–1479, Apr. 2020.
[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[24] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1222–1230.
[25] K. Li, W. Tao, and L. Liu, "Online semantic object segmentation for vision robot collected video," *IEEE Access*, vol. 7, pp. 107602–107615, 2019.
[26] K. Li, J. Zhang, and W. Tao, "Unsupervised co-segmentation for indefinite number of common foreground objects," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1898–1909, Apr. 2016.
[27] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[28] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[30] C. Liu, S. Li, F. Chang, and Y. Wang, "Machine vision based traffic sign detection methods: Review, analyses and perspectives," *IEEE Access*, vol. 7, pp. 86578–86596, 2019.
[31] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*. [Online]. Available: http://arxiv.org/abs/1512.02325
[32] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[33] Z. Liu, J. Du, F. Tian, and J. Wen, "MR-CNN: A multi-scale region-based convolutional neural network for small traffic sign recognition," *IEEE Access*, vol. 7, pp. 57120–57128, 2019.
[34] J. Noh, W. Bae, W. Lee, J. Seo, and G. Kim, "Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9725–9734.
[35] C. Premachandra, S. Ueda, and Y. Suzuki, "Detection and tracking of moving objects at road intersections using a 360-degree camera for driver assistance and automated driving," *IEEE Access*, vol. 8, pp. 135652–135660, 2020.
[36] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[37] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.
[38] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767
[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[41] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229
[42] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 2809–2813.
[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[44] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.
[45] W. Tao, "Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation," *Opt. Eng.*, vol. 46, no. 12, Dec. 2007, Art. no. 127002.
[46] W. Tao, H. Jin, and Y. Zhang, "Color image segmentation based on mean shift and normalized cuts," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 37, no. 5, pp. 1382–1389, Oct. 2007.
[47] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2001, p. 1.
[48] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.
[49] G. Wang, Z. Xiong, D. Liu, and C. Luo, "Cascade mask generation framework for fast small object detection," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2018, pp. 1–6.
[50] Y. Wang, L. Wang, Y. H. Hu, and J. Qiu, "RailNet: A segmentation network for railroad detection," *IEEE Access*, vol. 7, pp. 143772–143779, 2019.
[51] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1572–1583, Apr. 2020.
[52] B. Yan, N. Xu, G. Wang, S. Yang, and L. P. Xu, "Detection of multiple maneuvering extended targets by three-dimensional Hough transform and multiple hypothesis tracking," *IEEE Access*, vol. 7, pp. 80717–80732, 2019.
[53] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition using spatially weighted HOG trees," in *Proc. 15th Int. Conf. Adv. Robot. (ICAR)*, Jun. 2011, pp. 61–66.
[54] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4203–4212.
[55] Z. Zhang and W. Tao, "Pedestrian detection in binocular stereo sequence based on appearance consistency," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1772–1785, Sep. 2016.
[56] Z. Zhang, W. Tao, K. Sun, W. Hu, and L. Yao, "Pedestrian detection aided by fusion of binocular information," *Pattern Recognit.*, vol. 60, pp. 227–238, Dec. 2016.
[57] Y. Zhu, M. Liao, M. Yang, and W. Liu, "Cascaded segmentation-detection networks for text-based traffic sign detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 209–219, Jan. 2018.
[58] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.
[59] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.
[60] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019, *arXiv:1905.05055*. [Online]. Available: http://arxiv.org/abs/1905.05055

LIMAN LIU received the B.S. degree in biomedical engineering and the M.S. degree in electronics and information engineering from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2002 and 2005, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute for Pattern Recognition and Artificial Intelligence, HUST, in 2012. From 2006 to 2009, she was with O2Micro Corporation. She is currently an Associate Professor with the School of Biomedical Engineering, South-Central University for Nationalities, Wuhan. Her research interests include signal processing, mobile communications, image processing, and computer vision. She has authored or coauthored more than ten articles in image processing and object recognition. Her works have been cited beyond 300 times. She has received the Excellent Graduate Award from HUST, in 2002 and 2005.



YUNTAO WANG received the B.S. degree in electronic information science and technology from Hubei Engineering University, Xiaogan, China, in 2018. He is currently pursuing the master's degree in biomedical engineering with the School of Biomedical Engineering, South-Central University for Nationalities, Wuhan, China. His research interests include image processing and object recognition.

KUNQIAN LI (Member, IEEE) received the B.S. degree from the China University of Petroleum (UPC), Qingdao, China, in 2012, and the Ph.D. degree from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2018. He is currently a Lecturer with the College of Engineering, Ocean University of China, Qingdao. His research interests include image segmentation and object recognition. He serves as a reviewer for many journals, such as the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and *BMC Bioinformatics*.

JIE LI received the B.S. and M.S. degrees from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 2012 and 2019, respectively. His research interests include object detection and deep learning.