IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 23, 2020, accepted December 22, 2020, date of publication December 25, 2020,
date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047414

Traffic Sign Detection and Recognition Using Multi-Scale Fusion and Prime Sample Attention

JINGHAO CAO, JUNJU ZHANG, AND WEI HUANG
School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
Corresponding author: Junju Zhang (junjuzhang@njust.edu.cn)

ABSTRACT Traffic sign detection, though one of the key technologies in intelligent transportation, still has bottleneck in accuracy due to the small size and diversity of traffic signs. To solve this problem, we proposed a two-stage CNN object detection algorithm based on multi-scale feature fusion and prime sample attention. We improved the original Faster R-cnn model in terms of feature extraction and sampling strategy. For feature extraction, to elevate the ability of neural networks to detect small objects, we adopted HRNet as the feature extractor. There are four stages in HRNet - a series of high resolution subnets as the starting point with repeated adding parallel high to low resolution subnets to form other stages. In the whole process, the information in the parallel multi-resolution sub-network is repeatedly exchanged to perform repeated multi-scale fusion. For sampling strategy, we adopted a simple and effective sampling and learning strategy called Prime Sample Attention (PISA), consisting of Importance-based Sample Reweighting (ISR) and Classification Aware Regression Loss (CARL). PISA proposed the concepts of IoU Hierarchical Partial Sorting (IoU-HLR) and Hierarchical Partial Score Sorting (Score-HLR), which sort the importance of positive samples and negative samples in mini-batch respectively. With the proposed method, the training process is focusing on prime samples rather than evenly treat all ones. The algorithm complexity of our method is lower than that of other state-of-the-art. After experiments by TT100K dataset, our method can attain a comparable or even better detection accuracy and robustness.

INDEX TERMS Traffic sign detection, multi-scale, prime sample attention, features extract.

I. INTRODUCTION

In recent years, technology of road traffic signs detection has attracted wide attention due to increasing cases of traffic accidents resulting from ignorance of road signs. Not only the academia has conducted in-depth research, but also BMW, Mercedes-Benz and other well-known automobile companies have invested in business plans to study the technology, BMW Road Environment Perception System (REPS) as an example. The REPS system includes detection of front cars, pedestrians, and the traffic signs. Those studies realized the automatic detection and recognition of traffic signs through computer vision technology; however, the accuracy of detection needs improving.

Using reflective materials, solid colors and simple geometric signs made the traffic signs eye-catching; however, it remains difficult to detect and identify traffic signs by computer due to the unstable features of traffic signs in different occasions, such as viewing angle changes, self-damage, and bad weather. Accurate identification and detection of traffic signs remains a challenge.

For researchers, there are several technical challenges in achieving high accuracy of detection. Firstly, it is always difficult for computers to detect relatively small objects in the entire image.

Secondly, traffic signs of multiple instructions in a fixed shape result in the difficulty of accurate detection. For example, the traffic signs in the dataset TT100K [1] are in three shapes: rectangle, triangle, and circle, but they fall into 200 types of instruction.

Additionally, the accuracy of detection may be affected by multiple factors, such as the fluctuation of the object size in the field of view, bad weather, and damage to the traffic sign itself. Fig.1 shows some samples of detection difficulty.

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

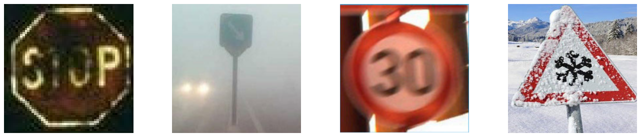

(a) (b) (c) (d)

FIGURE 1. Samples of traffic signs difficult to detect: (a) damaged and unideal light; (b) fog; (c) motion blur; (d) snow.

Aiming at these problems, a traffic sign detection system based on improved Faster R-CNN [2] neural network is designed. Our detection model accomplishes comparable detection and classification accuracy with state-of-the-art method. The main contributions of this article are as follows:

First, we innovatively adopt HRNet [3], [4] as the feature extractor for traffic sign object detection. For the detection of small objects, the multi-scale fusion feature extraction layer retains more information than the concatenated feature extraction layer. HRNet can maintain high-resolution representation throughout the process, starting with a high-resolution subnet in the first stage, adding subnets from high to low resolution one by one to form more stages, connecting the multi-resolution subnets in parallel, and exchanging the information in the parallel multi-resolution subnets repeatedly throughout the process to perform repeated multi-scale fusion. This multi-scale fusion method is more advanced than traditional methods. Moreover, for traffic signs with fewer shapes but more types, high resolution feature map can provide neural networks with more effective information

Second, we adopt the PISA [5] (Prime Sample Attention) method to optimize our model. PISA is a simple and effective sampling strategy through a simple weighting scheme to make the neural network focus on the samples which impose a greater impact on the training result (AP). Consequently, the learning efficiency of neural network gets higher, and the detection accuracy and robustness are enhanced.

The model designed in this paper shows a significant superiority compared to state-of-the-art in the dataset TT100K (Tsinghua-Tencent 100K). The rest of this paper is organized as follows. Section 2 briefly reviews the related work in recent computer vision approaches for object detection and small object detection. Section 3 presents the proposed Faster R-cnn approach for traffic signs detection. Section 4 discusses our results and ablation researches. Section 5 concludes our work.

II. RELATED WORK

Traditional image detection technology is based on manually extracting image features, such as SIFT (Scale Invariant Feature Transform) [6], SURF (Speeded-Up Robust Features) [7], and HOG [8] (Histograms of Oriented Gradient).

SIFT(Scale Invariant Feature Transformation) approach proposed by David G. Lowe, combined with local spatial histogramming and normalization, performed very well in object detection and image matching. Bay *et al.* proposed the SURF (Speeded-Up Robust Features) algorithm. It solves the shortcomings of SIFT calculation complexity and time-consuming on the basis of maintaining the excellent performance of SIFT, because the extraction of interest points and the description of feature vectors are improved, and the calculation speed is increased. Dalal *et al.* proposed the Histograms of Oriented Gradient descriptors with a conventional SVM [9] based sliding windows classifier, which method obtained good performance in human detection.

Traditional object detection algorithms are still widely used for fast calculation speed and low memory footprint [10]–[13]. For example, Anant Ram Dubey [14] *et al.* used HOG-SVM method to detect road objects, and Takaki Masanari [15] used SIFT [6] method to detect traffic signs. However, the accuracy of traditional object detection algorithms cannot compete with intelligent algorithms.

With the development of computer vision technology, machine learning and deep learning algorithms have been widely used in object detection with their high detection accuracy [16]–[20]. Deep learning algorithms can independently train and learn network models based on the labeled object dataset.

Deep learning detection algorithms include two-stage and single-stage algorithms. The two-stage algorithms include R-cnn [21], Fast-Rcnn [22], Faster R-cnn [2], R-FCN [23], and Mask R-cnn [24], etc. Single-stage object detection algorithms are represented by SSD [25], YOLO [26]–[28], etc.. Such algorithms directly predict objects' location and category without region proposal. However, the single-stage algorithm is not as accurate as the two-stage algorithm, especially in the detection of small objects.

Faster R-cnn [2] is a two-stage object detection algorithm proposed by He *et al.* in 2015. It mainly includes feature extractor, Region Proposal Network, ROI pooling, and Fully Connected Layers to classification and regression. Because of its excellent performance in object detection tasks, it is widely used in face, vehicle, pedestrian, traffic sign detection and other fields.

The focus of research on the traffic sign detection based on deep learning algorithm is to improve the feature extraction and sampling strategy of convolutional neural networks.

In response to this problem, Wang et al [29] improved the feature extractor of Casade R-cnn [30]. They adopted Resnet101 [31] as the backbone of Casade R-cnn, but their model is too complex to achieve real-time video detection. Han et al [32] improved the feature extractor and sampling strategy of Faster R-cnn [2]. They tried to use the shallow layer of VGG16 [33] as the feature map of RPN, and adopted OHEM [34] to improve the sampling strategy of their model. But this would lose the semantic information of the deep feature map of VGG16, and reduce the robustness of the model, and OHEM does not significantly improve performance. Jiang et al [35] improved the loss function of Yolov3 [28]. They adopted GIoU [36] and Focal Loss [37] as the loss function of the model, but the detection accuracy

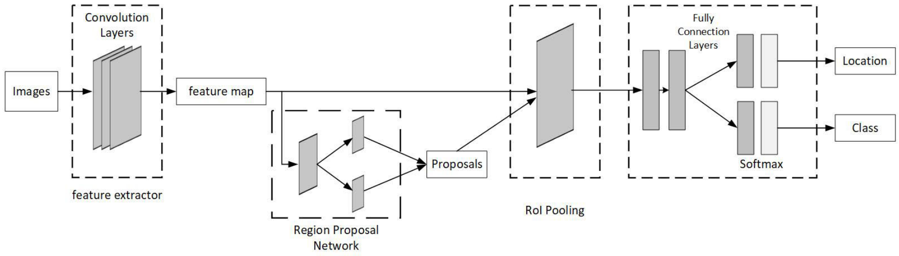


FIGURE 2. The Architecture of Faster R-CNN.

of the single-stage detection algorithm was too low to meet actual needs.

Based on the above related work, this paper innovatively adopted HRNet [3], [4] and HRFPN [3], [4] to improve the feature extractor of Faster R-cnn [2], and adopted PISA [5] strategy to optimize the learning strategy of Faster R-cnn. Our method can attain a comparable or even better detection accuracy and robustness than many state-of-the-art methods.

III. APPROACH

The architecture of Faster R-cnn is shown in Fig. 2. The image input is downsampled by the feature extractor to get the feature map. After the feature map fed into the Region Proposal Network, several proposals are obtained. These proposals are fed into the Roi Pooling Layer with the feature map to obtain that with the proposals, which is then used in the Prediction Layer. The Classification Layer predicts the category of proposals, and in the meanwhile obtains the precise position of the objects through bounding box regression.

A. EXTRACTOR

Traditional feature extraction backbones like VGG16 and ResNet have poor performance in the detection of traffic signs. One of the reasons is that they only make region proposals based on the last feature map of the extractor, but the receptive field of feature map is too wild. Taking Faster R-cnn as an example, if VGG16 is used as the extractor, the theoretical receptive field of the feature map output by RPN network is 228×228; if ResNet50 is used as an extractor, the theoretical receptive field is 299 × 299. We adopted HFM (Hot Feature Map) [38] as the visualization of the feature map. The calculation formula of HFM can be expressed as (1):

$$HFM = \sum_{c=0}^{C} feature\_map(c, height, width) \tag{1}$$

Fig. 3 shows a sample of traffic signs dataset with a size of 800 × 800×3. Fig. 4 shows the visualized feature maps of Fig. 3, the feature map generated by VGG16 and ResNet50. Since the human eye is much more sensitive to color images than grayscale images, we map the grayscale picture of the hot feature map to YB color space in Fig. 4.

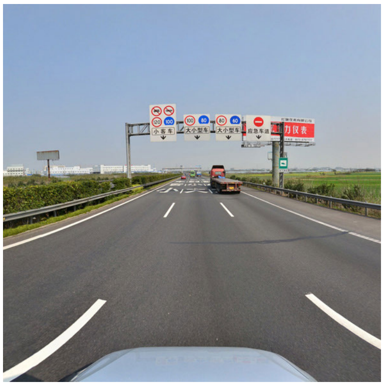
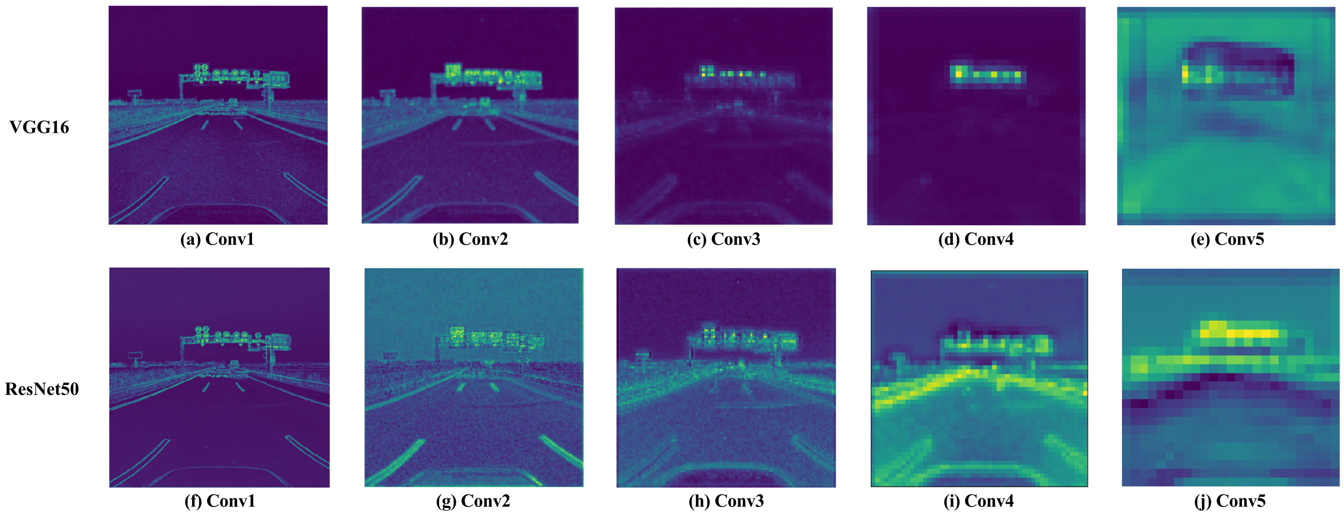


FIGURE 3. A sample of traffic signs dataset with a size of 800 × 800×3.

It can be seen from Fig. 4 that the resolution of the feature map in deep layer is significantly reduced. Although feature map in deep layer contains rich semantic information of the image, it loses some detailed information of the object, which will significantly reduce the performance of small objects detection by the neural network. In other words, the large receptive field feature extractor is not suitable for the detection of small objects. In response to the above problems, we found that the extractor, which is mainly used for human pose estimation, has an amazing effect on the detection of small objects. Because the multi-scale fusion of feature maps, for example, FPN, can significantly improve the ability of neural networks to detect small objects. The structure of HRNet is shown in Fig. 5. The backbone of HRNet include four stages, the network started with a series of high resolution convolution layers, then repeatedly adding and connecting the parallel multi-resolution subnets to form the 2nd, 3rd,4th stages. In the whole process, the information in the parallel multi-resolution convolutional layer is repeatedly exchanged to perform repeated multi-scale fusion.

The specific process of fusion in the HRNet is shown in Fig. 6. In the backbone of HRNet, the layer with the same resolution is directly copied to the next layer. Bilinear upsample is used to upsample the low-resolution feature layer, and then use 1 × 1 convolution layer to match the channels of high resolution layer. For the high-resolution feature layer, we adopt 3 ×3 stride convolution kernel to downsample. After completing the upsample and downsample process,



(a) Conv1 (b) Conv2 (c) Conv3 (d) Conv4 (e) Conv5

(f) Conv1 (g) Conv2 (h) Conv3 (i) Conv4 (j) Conv5

FIGURE 4. **False color images of HFMs generated by VGG16 and ResNet50,, The (b)–(f) respectively come from the five different stages of VGG16, the (g)-(k) come from the five different stages of ResNet50.Their panel size are Conv1: 400 × 400; Conv2: 200 × 200; Conv3: 100 × 100; Conv4: 50 × 50; Conv5: 25 × 25. all of them has been resized to the same size for convenient display.**

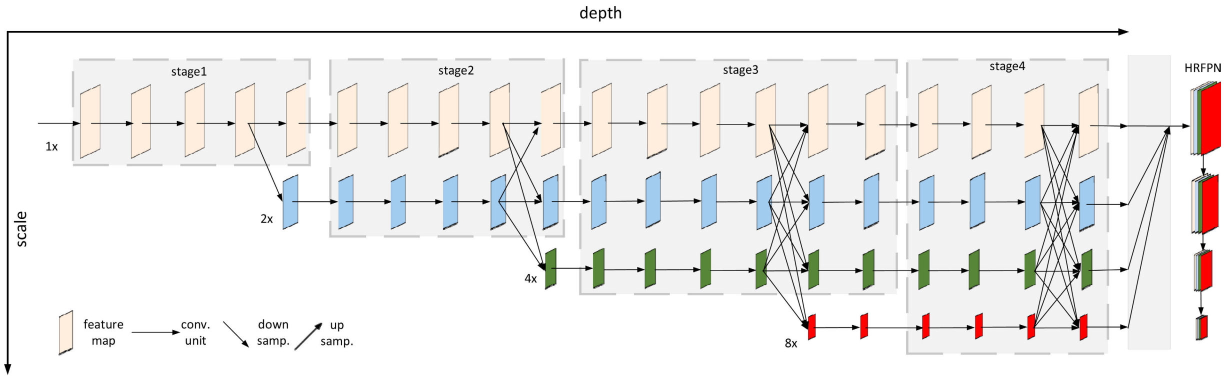
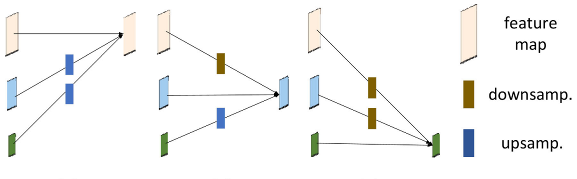


FIGURE 5. **The architecture of HRNet backbone and HRPFN.**



(a) (b) (c)

FIGURE 6. **The specific process of fusion in the HRNet network. The layer with the same resolution is directly copied to the next layer. We used bilinear upsample method and 3 × 3 convolution kernel to dawnsample.**

feature maps of different resolutions will be fused in the form of element-add. In order to reduce the information loss in the downsample process, pooling layer is not used. We adopted a feature pyramid network based on HRNet – HRFPN - to enhance the neural network's ability to detect small objects. Its architecture is shown in Fig. 5. It mixes the output representations, from all the four resolutions through a 1 × 1 convolution, and produce a 15x-dimensional representation, and then reduce the dimension of the high-resolution representation to 256, similar to FPN [39].

Ke Sun et. al proposed in their HRNet paper three models of w18, w32, and w48. Among them, 18, 32, and 48 represent the channel number of the last layer of feature layers. We adopted w18 as the improved feature extractor of Faster R-cnn. The reasons for this choice will be explained in the ablation research. We resized the images with a size of 3 × 2048×2048 into 3 × 800×800 and fed them into HRNet, then got 18 × 200×200, 36 × 100×100, 72 × 50×50 and 144 × 25×25 feature maps. Then we fed them into HRFPN, unified the channels of these feature maps to 256 through 1 × 1 convolution kernel, and fused them to obtain 256 × 200×200 feature maps, and then got 256 × 100 ×100, 256 × 50 × 50, 256 × 25 × 25 and 256 × 13 × 13 feature maps through average pooling layer. They are sent to RoI Pooling Layer separately.

Compared to the traditional sequential top-down fusion strategy, HRNet can maintain high resolution instead of restoring resolution from low to high. It performs repeated multi-scale fusion with the help of low-resolution block of the same depth and similar level to improve

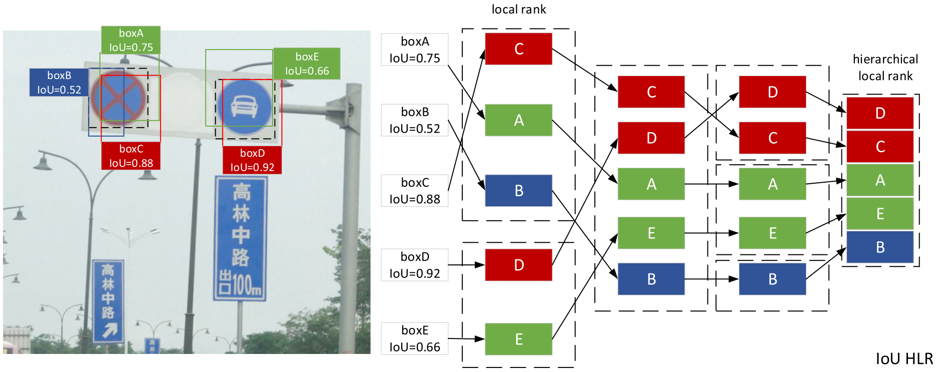


FIGURE 7. Steps to compute IoU-HLR. First divide all samples into different groups according to their nearest groundtruth object. Next, the samples in each group are sorted using IoU descending order with groundtruth. Subsequently, samples are taken with the same IoU-LR and sorted in descending order.

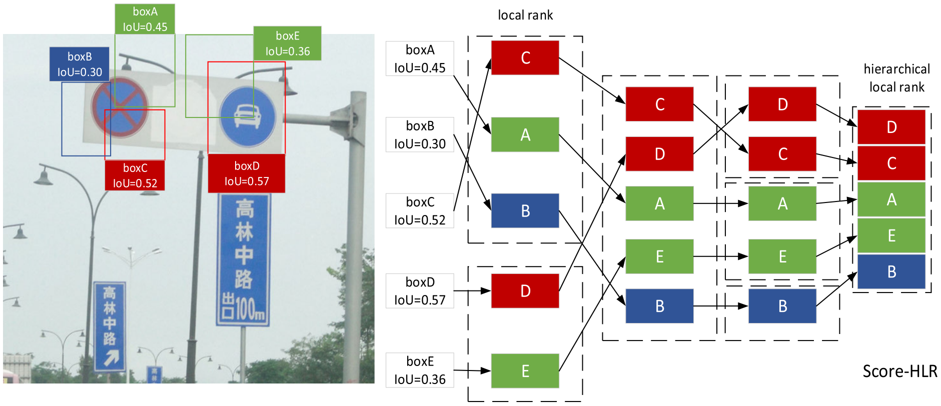


FIGURE 8. Steps to compute Score-HLR. Use the highest score in all foreground categories as the score of the negative sample, and then perform the same steps as computing IoU-HLR.

high resolution rate, so the feature map may be more accurate.

B. SAMPLING STRATEGY

The sampling process of original RPN is to randomly select some positive and negative samples from all anchors. But according to Ke Sun *et al.* [4], the samples in each mini-batch are neither independent nor equally important. Therefore, we adopted a simple and effective sampling and learning strategy called Prime Sample Attention (PISA), which shifts the focus of the training process to prime samples. Our experiments showed that focusing on prime samples is usually more effective than on hard and random samples when training the detection neural network. According to Ke Sun *et al.* [4], the positive samples that affect training are mainly those with higher IoU, while the negative with higher classification scores.

PISA proposed the concepts of IoU Hierarchical Partial Sorting (IoU-HLR) and Hierarchical Partial Score Sorting (Score-HLR), which make model sort the importance of positive and negative samples respectively after region proposal in each iteration. As shown in Fig. 7, to compute IoU-HLR, we first divided all samples into different groups according to their nearest groundtruth object. Next, the samples in each group are sorted using IoU descending order with groundtruth, and then the IoU local ranking (IoU-LR) is obtained. Subsequently, samples are taken with the same IoU-LR and sorted in descending order. Specifically, we collected and classified all top1 IoU-LR samples, followed by top2, top3, and so on. These two steps were followed to sort all samples.

As shown in Fig. 8, we computed the Score-HLR of negative samples in a similar way to IoU-HLR. Unlike the positive samples that are naturally grouped by each gt object, negative ones may appear in the background area. So, we grouped them into different clusters based on NMS first. Then we chose the highest score in all foreground categories as that of the negative sample, and then perform the same steps as computing IoU-HLR.

PISA consists of two components: Importance-based Sample Reweighting (ISR) and Classification Aware Regression Loss (CARL). With the proposed method, the training process is focusing on prime samples rather than evenly treating all ones.

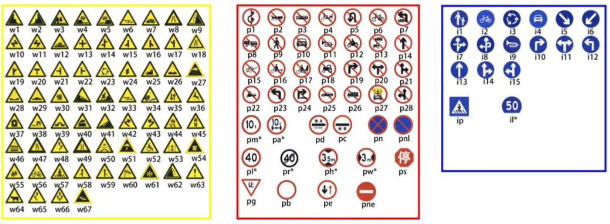

w1 w2 w3 w4 w5 w6 w7 w8 w9
w10 w11 w12 w13 w14 w15 w16 w17 w18
w19 w20 w21 w22 w23 w24 w25 w26 w27
w28 w29 w30 w31 w32 w33 w34 w35 w36
w37 w38 w39 w40 w41 w42 w43 w44 w45
w46 w47 w48 w49 w50 w51 w52 w53 w54
w55 w56 w57 w58 w59 w60 w61 w62 w63
w64 w65 w66 w67
p1 p2 p3 p4 p5 p6 p7
p8 p9 p10 p11 p12 p13 p14
p15 p16 p17 p18 p19 p20 p21
p22 p23 p24 p25 p26 p27 p28
pm* pa* pd pc pn pnl
pl* pr* ph* pw* ps
pg pb pe pne
i1 i2 i3 i4 i5 i6
i7 i8 i9 i10 i11 i12
i13 i14 i15
ip il*

FIGURE 9. Some categories in the TT100K dataset.

As described by Yuhang Cao *et al.* [5], the computation of ISR can be expressed as (2)-(4). Firstly, we ranked the samples by IoU-HLR or Score-HLR, and then transformed this rank to a real value, this process can be expressed as (2)

$$u_i = \frac{n_{\max} - r_i}{n_{\max}} \tag{2}$$

u_i is the importance value of the i_{th} sample of category j. $n_{\max}$ is the maximum value of n_j in all categories, which ensures that samples in the same order of different categories will be assigned the same u_i.

And then we need a monotonically increasing function to further increase sample importance value u_i to a loss weight w_i. Among them, γ is a degree factor indicating the to, and β is the bias that determines the minimum sample weight. which important samples will be prioritized

$$w_i = ((1-\beta)u_i + \beta)^{\gamma} \tag{3}$$

Based on the above improvements, the classification loss of Faster R-cnn can be rewritten as (4), where CE is the abbreviation of cross entropy; s and $\hat{s}$ represent the prediction score and classification object; n and m are the number of positive samples and negative samples respectively. and In order to keep the total loss relatively stable,we normalized w to w'.

$$\begin{aligned}
L_{cls} &= \sum_{i=1}^{n} w_i' CE(s_i, \hat{s}_i) + \sum_{j=1}^{m} w_j' CE(s_j, \hat{s}_j) \\
w_i' &= w_i \frac{\sum_{i=1}^{n} CE(s_i, \hat{s}_i)}{\sum_{i=1}^{n} w_i CE(s_i, \hat{s}_i)} \\
w_j' &= w_j \frac{\sum_{j=1}^{m} CE(s_j, \hat{s}_j)}{\sum_{j=i}^{m} w_j CE(s_j, \hat{s}_j)}
\end{aligned} \tag{4}$$

The role of CARL is to highlight the prime samples, while suppressing other ones. CARL can optimize the process of localization and classification relevantly, its specific method can be expressed as (5)

$$\begin{aligned}
L_{carl} &= \sum_{i=1}^{n} c_i L(d_i, \hat{d}_i) \\
c_i &= \frac{v_i}{\frac{1}{n}\sum_{i=1}^{n} v_i} \\
v_i &= ((1-b)p_i + b)^k
\end{aligned} \tag{5}$$

p_i presents the predicted probability of the ground truth and d_i denotes the output regression offset. L is the commonly used smooth L1 loss.

With CARL, the classification branch can be supervised by regression loss, and the impacts of unprime samples are greatly suppressed, and the focus on the prime samples is strengthened..

IV. EXPERIMENTS

A. DATASET

TT100K [1], provided by Tsinghua University and Tencent Corporation, is a large traffic-sign benchmark from 100000 Tencent Street View panoramas. The dataset contains 9176 images (6105 for training and 3071 for testing). These images contain 221 types of traffic signs, and cover large variations in illuminance and weather conditions with a size of 2048 × 2048. Each traffic-sign in the benchmark is annotated with a class label, its gt bbox (ground truth bounding box) and pixel mask.

TABLE 1. Statistical table of TT100K.

Name	Tsinghua-Tencent 100K
Images	9176 (6105 for training, 3071 for testing)
Gt Bboxes	16527
Categories	227
Area (Pixels)	2048×2048
Range of gt bbox aspect	(0.34, 1.38)

We performed statistical analysis on the TT100K dataset, and summarized the statistical results in Table 1. From Fig. 10(a) and Fig. 10(b), we can conclude that the area of the gt bboxes in the TT100K dataset is mostly less than 9216 pixels, accounting for 92.81%. Among the gt bboxes, those with an area ranging from 1024 pixels to 9216 pixels account for 53.42%, and those with an area less than 1024 pixels account for 39.28%. Although some dataset like COCO [41] divides the objects into three groups based on their size, namely, small(area∈[1,1024)), middle(area∈[1024,9216)), and large (area>=9216), we think this method unsuitable for TT100K, because the size of image in TT100K is 2048 × 2048. So, even if the area of a gt bbox is 9216 pixels, it only occupies about 0.22% of the entire image, which obviously cannot be called a "large object". From the perspective of practical applications, we have retained all 221 types of objects in TT100K for detection.

In addition, Figure 10(C) is a scatter diagram of gt bbox. We used least squares regression to estimate the approximate interval of the aspect ratio of the bbox to (0.34, 1.38). We hoped to use the small area and aspect ratio of the anchor in the RPN of Faster R-cnn, thereby improving the detection accuracy of the algorithm. Unfortunately, such optimization has little effect. The AP (Average Precision) is 0.237 when obtained by original Faster_R-cnn, 0.238 by Faster R-cnn with a suitable aspect ratio and 0.242 by Faster_R-cnn with a

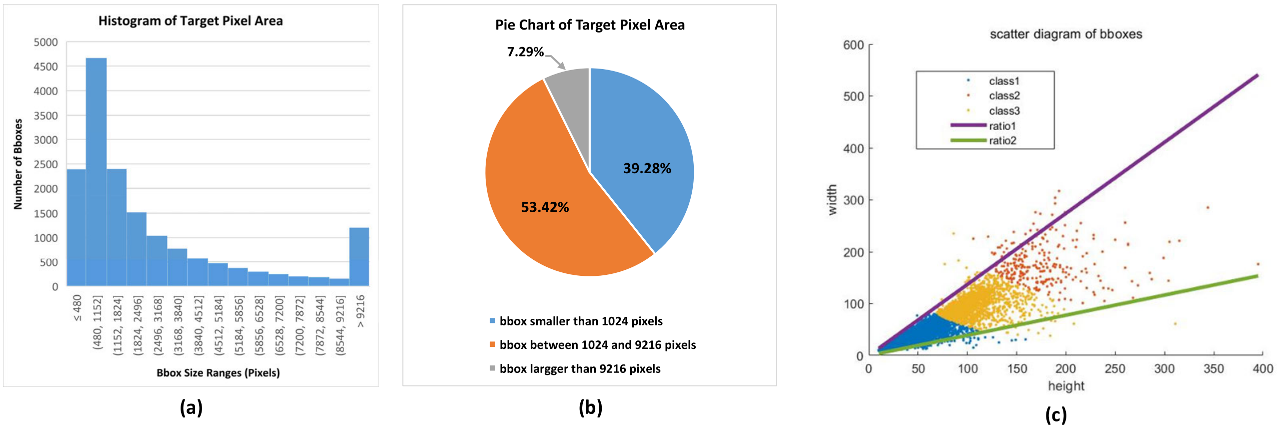


FIGURE 10. Statistics about the TT100K dataset. (a) is the histogram of object pixel area, (b) is the pie chart of object pixel area, it reflects the percentage of area(pixels) of the gt bbox, (c) is scatter diagram of ground truth bboxes and the approximate range of their aspect ratio.

TABLE 2. Comparison of their model complexity and computational efficiency.

	Backbone	Neck	RoI Head	GFLOPs	Params	Inference speed
1: ours	HR_w18	HRFPN	PISA	123.28	40.08M	26FPS
2. Wang et al.	Res101	FPN	Rand	210.41	88.62M	14FPS
3. Han et al.	VGG16	\	OHEM	120.53	39.37M	21FPS
4. Jiang et al.	Dark53	\	Rand	124.31	62.71M	43FPS
5. CRCNN	Res50	FPN	Rand	162.86	64.65M	17FPS
6. FRCNN	Res50	FPN	Rand	135.51	42.25M	24FPS

suitable aspects ratio and smaller anchor size. The detection performance was not improved significantly.

B. TRAINING DETAILS

The experimental environment of our approach is NVIDIA TITAN XP graphics card (if not specified, our experiment is usually implemented by two graphics cards working in parallel), Ubuntu16.04LTS system, CUDA10.0, and Pytorch1.3.1 programming framework based on Python 3.7.2.

In the preprocessing of the dataset, we first resized the input image to 800 × 800, then we used the randomflip strategy to augment the dataset. In each epoch of training, the probability of an image in training dataset being randomly flipped is 0.5.

During the training process, the total epoch is 48, and the initial learning rate is 0.02. We used the "linear warmup [31]" method to slowly increase the learning rate to 0.02. The warm up iterations is 500, and the warm up ratio is 0.001. The decay ratio is 0.1. The learning rate will be reduced to 0.002 after 32 epochs, and to 0.0002 after 44 epochs. We used the "momentum" method to accelerate the gradient descent, the momentum coefficient is 0.9, and we used the "weight decay" method in order to prevent overfitting. The weight decay coefficient is 0.0001. [40]

In addition, for (3) and (5), we adopted the conclusions of the original paper after ablation study, where γ_{p} =2.0, γ_{n} =0.5, $\beta_{\mathrm{p}} = \beta_{\mathrm{n}}$ =0,k=1,b=0.2, where γ, β for ISR. Among them, γ_{p} and β_{p} are the weight and bias when ranking positive samples, while γ_{n}, β_{n} are the weight and bias when ranking negative samples. k and b for CARL. The specific experimental details will be demonstrated in the ablation study.

C. DETECTION PERFORMANCE AND EFFICIENCY

We evaluated traffic sign detection methods from the aspects of algorithm complexity, computing speed, accuracy, and robustness. We compared our method with two representative generic object detectors Faster-RCNN-FPN and Casade R-cnn, and three state-of-the-art traffic sign detectors proposed by Wang *et al.*, Han *et al.* and Jiang *et al.*. The experimental results in the second section showed that the AP obtained by the original Faster R-cnn is only 0.237, which is much lower than other states of the arts, so we used the improved Faster R-cnn, namely Faster R-cnn with FPN, instead of the traditional one to compare with our method.

We used the number of parameters, FLOPs (Floating point operations) and Inference speed to represent the algorithm complexity and computational efficiency of these methods. The results are shown in Table 2.

The parameters of the model we built is 40.08M, of which HRNet is 21.3M, accounting for about 53%. It is 4.2M less than ResNet50 (25.5M Params), but higher than VGG16 (14.7M Params). However, the performance of HRNet is better than the classic feature extractor. Our model has fewer

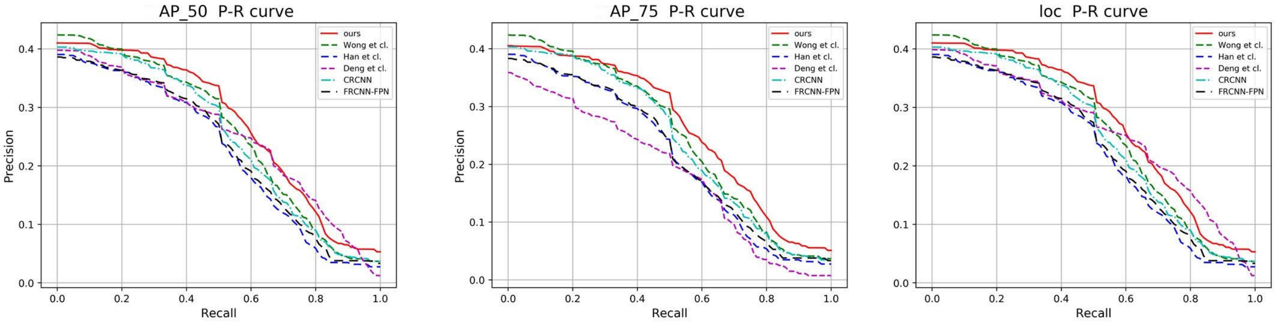


FIGURE 11. Precision-Recall curves for AP_{50}, AP_{75}, Loc of six object detection methods.

parameters and FLOPs than the models of Wang *et al.* simplified VGG16 network. This will reduce the robustness of the model. In addition, the inference speed of our model has reached 26FPS. Although compared to single-stage algorithms like SSD and YOLO, the speed of our method is slower, when compared to two-stage algorithms like Faster R-cnn and Casade R-cnn, the inference speed of our model is still better than most traditional two-stage algorithms.

We use AP (AP, AP_{50}, AP_{75}) and AR(Average Recall) to evaluate the accuracy of methods. The results are shown in Table 3.

TABLE 3. Comparison of detection performance of different methods.

	AP	AP_{50}	AP_{75}	AR
1: ours	**0.352**	**0.444**	**0.428**	**0.450**
2. Wang et al.	0.336	0.418	0.404	0.423
3. Han et al.	0.292	0.363	0.348	0.369
4. Jiang et al.	0.264	0.409	0.302	0.305
5. CRCNN	0.327	0.401	0.389	0.402
6. FRCNN	0.299	0.373	0.354	0.371

In all experiments, we uniformly used 800 × 800 images as input images, except that the loss function of Jiang *et al.* (improved YOLOv3) converges slowly, and AP is relatively stable after 273 epochs. We trained the rest of the models 48 epochs, where the learning rate decay steps are 32 and 44. The AP obtained by our method is 0.352, AP_{50} 0.444, AP_{75} 0.428. All the AP and AR obtained by the object detection algorithm are relatively low, because the dataset has 221 types. But, our detection accuracy is still 10%∼20% higher than the current state-of-the-art. For example, the AP obtained by our method is 18% higher than Faster R-cnn with FPN and 7% higher than Casade R-cnn.

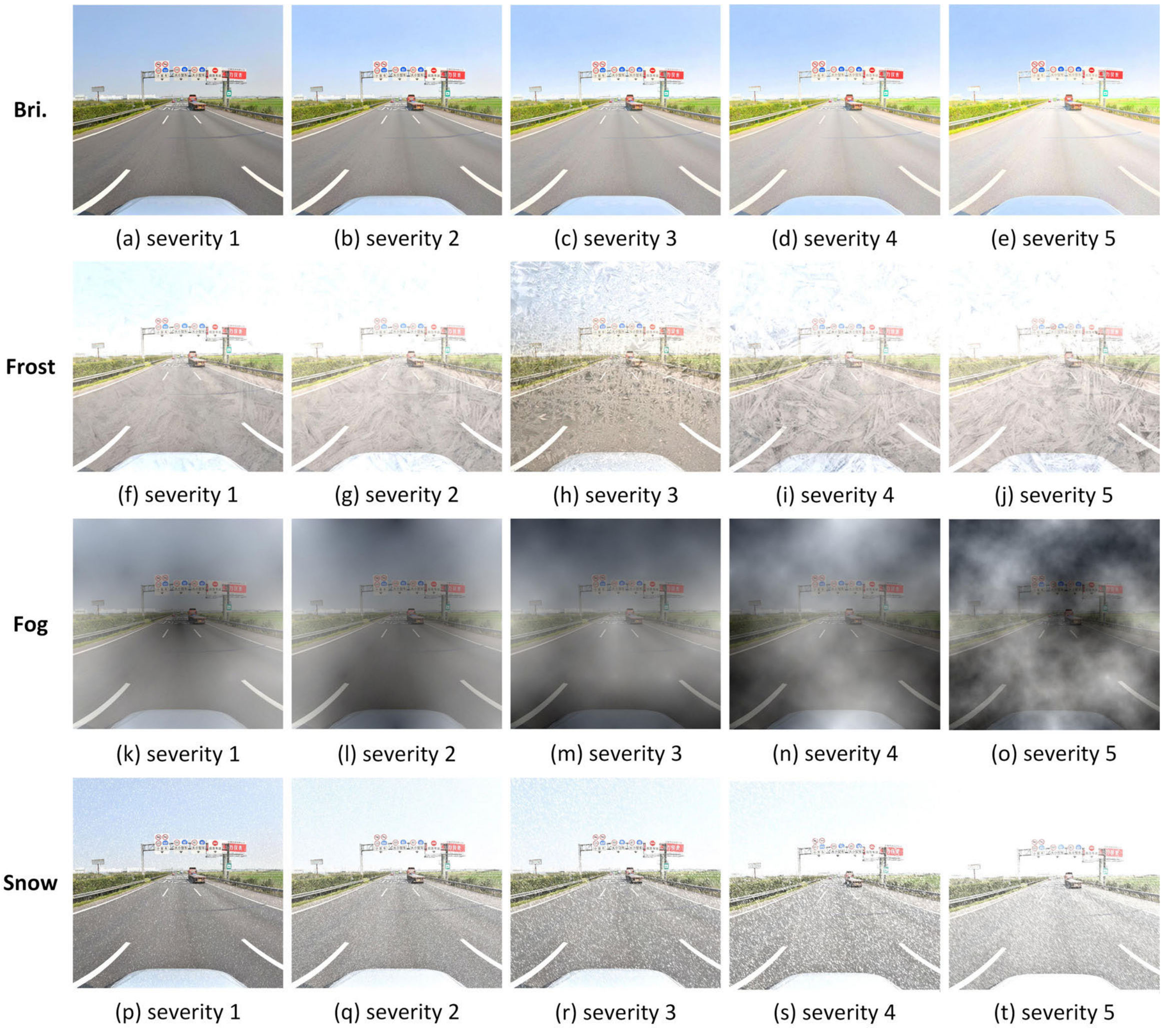
Fig. 11 illustrates the precision-recall curves of our method and the other methods of AP_{50}, AP_{75}, and Loc(localization errors ignored, but not duplicate detections). The precision-recall curve is a common measure to evaluate performance of object detectors. AP_{50} and AP_{75} are the APs when the IoU threshold is set to 0.5 and 0.75 respectively. Loc (localization errors ignored) is an indicator which consider classification accuracy only. They are all commonly used object detection method evaluation indicators. From Fig.11, the P-R curve of our method is generally high for other methods. Considering using a much lower resolution of 800 × 800, our method is still competitive from precision-recall curve perspective. The visualization of our detection results is shown in Fig.14. As can be seen from it, our method can effectively detect small and multiple objects in images.

In addition, we also evaluated the robustness of the model. We used Hendrycks and Dietterich's corruption image generation method [42] to simulate the four severe weather images under brightness, frost, fog, and snow, and divided each severe weather into 5 levels according to the benchmark in their paper [43]. For example, take Fig.3 as the original image, the generated corruption images are shown in Fig.12.

We considered the AP obtained by detecting the original dataset as "AP Clean", and that by detecting the corruption dataset as "AP Corruption", and used the percentage of "AP Corruption" in "AP Clean" to represent the method robustness. This process can be expressed by (6). The results of our robustness experiment are shown in Table 4.

$$Percentage = \frac{AP\ \ Corr.}{AP\ Clean} \tag{6}$$

In order to make the experimental results easy to observe, we plotted the AP percentage of different methods into a line chart as shown in Fig.13. It can be concluded from it that, in general, our method can maintain good performance for traffic sign detection under severe weather. For brightness, our algorithm can almost ignore the corruption of bad weather when its severity is low. When brightness severity is 1 and 2, the AP percentage is 99.43% and 95.74%, which is second only to Casade R-cnn with ResNet50. When the brightness severity is 3, 4, and 5, the performance is slightly inferior, but the AP percentage can still be maintained at 92.33%, 86.93%, and 79.83%, respectively. For frost, our method obtains AP percentages of 91.76%, 81.25%, 71.59%, 68.47%, and 61.36%, respectively. In our experiment, when severity is 1, the AP Percentage is second only to Wang. *et al.*. When severity is 2, the AP percentage is the highest. When severity is 3, the AP percentage is only lower than Han *et al.*. When severity is above 3, the AP percentage is slightly inferior, but still higher than 60%, which is within the acceptable range.



(a) severity 1 (b) severity 2 (c) severity 3 (d) severity 4 (e) severity 5

(f) severity 1 (g) severity 2 (h) severity 3 (i) severity 4 (j) severity 5

(k) severity 1 (l) severity 2 (m) severity 3 (n) severity 4 (o) severity 5

(p) severity 1 (q) severity 2 (r) severity 3 (s) severity 4 (t) severity 5

FIGURE 12. Corruption image simulation of different severity. We simulated four kinds of severe weather, namely brightness, frost, fog, snow. Each weather is divided into 5 severity levels.

For fog, our method obtained 87.78%, 82.39%, 77.27%, 74.72%, and 67.90% as AP percentages. In our experiment, when severity is under 5, the AP percentage is second only to Faster R-cnn with RPN, and both reached more than 70%. When severity is 5, the AP percentage is slightly inferior, but it is still higher than 60%. For snow, our method obtained 86.08%, 67.90%, 64.20%, 52.84%, 46.59% as AP Percentages. Although in our experiment the AP percentage is always second only to Faster R-cnn with RPN, when severity is 5, our AP percentage is lower than 50%. It can be considered that our model has a slightly weaker detection ability for snowy weather and the best robustness for foggy weather.

D. ABLATION ANALYSIS

The main improvements in Faster R-cnn in this paper are feature extraction and sampling strategy. This section will discuss the impact of these two improvements in detail. The total epoch of our ablation experiment training is 48, and the learning rate decay steps are 32 and 44. The experimental results are shown in Table 5.

1) IMPROVED FEATURE EXTRACTOR

In the HRNet paper [3], Ke Sun provides three output sizes of the backbone, namely HRNet-w18, HRNet-w32 and HRNet-w40. Their final layer sizes are 18 × 18, 32 × 32, 40 × 40, respectively. The parameters of HRNet-w40 are 77.5M, but its detection performance of the COCO [41] dataset has not been significantly improved, so we do not consider it as a feature extractor of the traffic sign detection model. In Table 5, comparing experiments 1 and 4, 2 and 5, 3 and 6, when using the PISA sampling strategy, the AP obtained by using HRNet-w18 is 0.352 higher than HRNet-w32 by 0.007; when using the OHEM negative hard sample mining strategy, it is 0.319, which is lower by HRNET_w32 0.002; when the random strategy is adopted, it is 0.324, which is higher than HRNet-w32 by 0.004. In general, the performance of HRNet-w18 is better. Too many parameters for the low resolution input of 800 × 800 of HRNet-w32 result in overfitting. For experiments 2 and 5, we believe that the small size feature map caused the inability of OHEM.

TABLE 4. Comparison of robustness of detection methods.

	Bri.1		Bri.2		Bri.3		Bri.4		Bri.5	
	AP	%	AP	%	AP	%	AP	%	AP	%
1: ours	**0.350**	**99.43%**	**0.337**	**95.74%**	**0.325**	**92.33%**	**0.306**	**86.93%**	**0.281**	**79.83%**
2. Wang et al.	0.333	99.11%	0.31	92.26%	0.297	88.39%	0.282	83.93%	0.271	80.65%
3. Han et al.	0.290	99.32%	0.279	95.55%	0.275	94.18%	0.27	92.47%	0.266	91.10%
4. Jiang et al.	0.256	96.97%	0.246	93.18%	0.232	87.88%	0.218	82.58%	0.204	77.27%
5. Casade Rcnn	0.326	99.69%	0.317	96.94%	0.308	94.19%	0.296	90.52%	0.288	88.07%
6. Faster Rcnn FPN	0.292	97.66%	0.282	94.31%	0.277	92.64%	0.268	89.63%	0.222	74.25%
	Frost1		Frost2		Frost3		Frost4		Frost5	
	AP	%	AP	%	AP	%	AP	%	AP	%
1: ours	**0.323**	**91.76%**	**0.286**	**81.25%**	**0.252**	**71.59%**	**0.241**	**68.47%**	**0.216**	**61.36%**
2. Wang et al.	0.311	92.56%	0.272	80.95%	0.233	69.35%	0.226	67.26%	0.197	58.63%
3. Han et al.	0.258	88.36%	0.232	79.45%	0.214	73.29%	0.206	70.55%	0.190	65.07%
4. Jiang et al.	0.226	85.61%	0.197	74.62%	0.176	66.67%	0.175	66.29%	0.161	60.98%
5. Casade Rcnn	0.297	90.83%	0.255	77.98%	0.222	67.89%	0.212	64.83%	0.194	59.33%
6. Faster Rcnn FPN	0.269	89.97%	0.237	79.26%	0.206	68.90%	0.201	67.22%	0.183	61.20%
	Snow 1		Snow2		Snow3		Snow4		Snow5	
	AP	%	AP	%	AP	%	AP	%	AP	%
1: ours	**0.303**	**86.08%**	**0.239**	**67.90%**	**0.226**	**64.20%**	**0.186**	**52.84%**	**0.164**	**46.59%**
2. Wang et al.	0.244	72.62%	0.179	53.27%	0.17	50.60%	0.137	40.77%	0.117	34.82%
3. Han et al.	0.217	74.32%	0.158	54.11%	0.145	49.66%	0.121	41.44%	0.098	33.56%
4. Jiang et al.	0.244	92.42%	0.202	76.52%	0.195	73.86%	0.156	59.09%	0.143	54.17%
5. Casade Rcnn	0.239	73.09%	0.171	52.29%	0.165	50.46%	0.136	41.59%	0.11	33.64%
6. Faster Rcnn FPN	0.222	74.25%	0.164	54.85%	0.154	51.51%	0.127	42.47%	0.106	35.45%
	Fog1		Fog2		Fog3		Fog4		Fog5	
	AP	%	AP	%	AP	%	AP	%	fog5	
1: ours	**0.309**	**87.78%**	**0.29**	**82.39%**	**0.272**	**77.27%**	**0.263**	**74.72%**	**0.239**	**67.90%**
2. Wang et al.	0.277	82.44%	0.262	77.98%	0.244	72.62%	0.243	72.32%	0.221	65.77%
3. Han et al.	0.241	82.53%	0.227	77.74%	0.217	74.32%	0.211	72.26%	0.197	67.47%
4. Jiang et al.	0.22	83.33%	0.197	74.62%	0.177	68.18%	0.18	67.05%	0.162	61.36%
5. Casade Rcnn	0.267	81.65%	0.252	77.06%	0.242	74.01%	0.239	73.09%	0.218	66.67%
6. Faster Rcnn FPN	0.264	88.29%	0.249	83.28%	0.234	78.26%	0.235	78.60%	0.220	73.58%

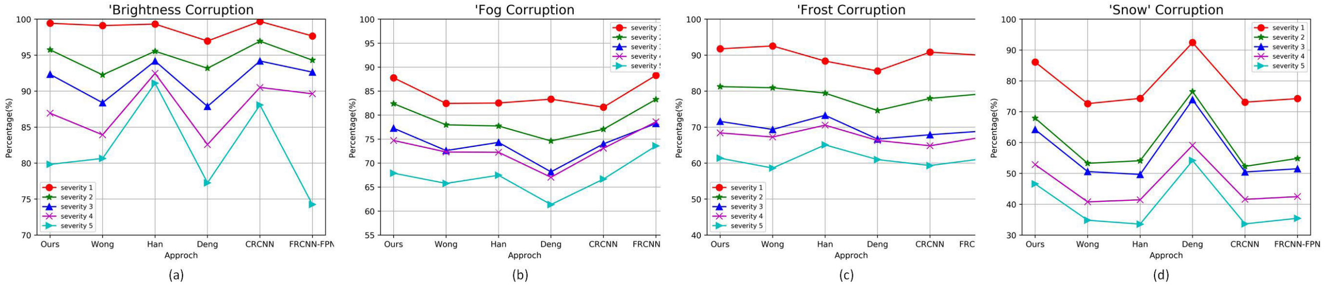


FIGURE 13. Comparison of AP percentage of corruption in different weather. (a) brightness corruption, (b) fog corruption, (c) frost corruption and (d) snow corruption.

2) SAMPLING STRATEGY

For HRNet-w18, the AP obtained by the PISA sampling strategy is 0.352, which is 0.033 and 0.028 higher than the OHEM and random strategies, respectively. For HRNet-w32, the AP obtained by the PISA sampling strategy is 0.345, which is 0.025 and 0.024 higher than the OHEM and random

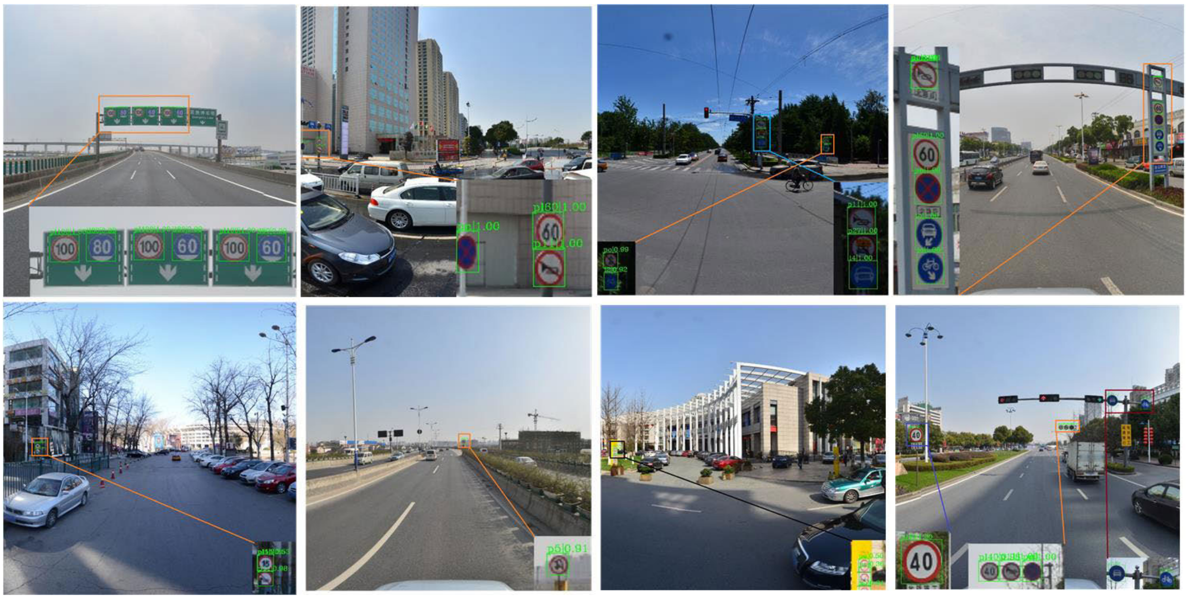


FIGURE 14. Visualization of the detection results attained by our method.

TABLE 5. Comparison of detection performance of different feature extractor and sampling strategies.

No.	Backbone	RoI Head	GFLOPs	Params	AP	AP50	AP75	AR
1	HRNet_w18	PISA	123.28	40.08M	**0.352**	**0.444**	**0.428**	**0.443**
2	HRNet_w18	Rand.	\	\	0.324	0.400	0.386	0.401
3	HRNet_w18	OHEM	\	\	0.319	0.393	0.382	0.394
4	HRNet_w32	PISA	184.46	59.98M	0.345	0.425	0.416	0.425
5	HRNet_w32	Rand.	\	\	0.320	0.393	0.384	0.391
6	HRNet_w32	OHEM	\	\	0.321	0.396	0.385	0.397

TABLE 6. Comparison of performance of different hyperparameters for PISA.

γp	βn	AP	γn	βp	AP	k	b	AP
0.5	0.0	0.349	0.5	0.0	**0.351**	0.5	0.0	0.350
1.0	0.0	0.350	1.0	0.0	0.350	1.0	0.0	0.351
2.0	0.0	**0.351**	2.0	0.0	0.351	2.0	0.0	N/A
2.0	0.1	0.350	0.5	0.1	0.350	1.0	0.1	0.351
2.0	0.2	0.348	0.5	0.2	0.350	1.0	0.2	**0.352**
2.0	0.3	0.349	0.5	0.3	0.349	1.0	0.3	0.350

strategies. respectively. This indicated that the PISA strategy is more suitable for traffic sign detection than other sampling strategies.

In addition, we also conducted ablation studies on the hyperparameters of PISA. The experimental results are shown in Table 6:

In Table 6, γ and β are for ISR. Among them, γ_p and β_p are the weight and bias when ranking positive samples, and γ_n, β_n when ranking negative samples. k and b are for CARL. The conclusion of the hyperparameter experiment is basically the same as the original paper [4]. So, we adopt $\gamma_\mathrm{P} = 2.0$, $\gamma_\mathrm{N} = 0.5$, $\beta_\mathrm{P} = \beta_\mathrm{N} = 0$ for ISR, and k = 1.0, b = 0.2 for CARL. According to the results in Section 2, the AP obtained by the original Faster R-cnn is 0.237, while the AP obtained by our model is 0.352, which is 0.115 higher, and an increase of about 48.5%.

V. CONCLUSION

In this paper, we proposed a two-stage CNN traffic sign detection algorithm based on improved Faster R-cnn. We used the parallel fusion feature extraction network, HRNet, to improve the feature extractor of Faster R-cnn and the attention mechanism of Faster R-cnn. Through the overall designs, the algorithm complexity of our method is lower than that of other state-of-the-art. After experiments by TT100K dataset, our method can attain a comparable or even better detection accuracy and robustness. In the future, we will continue to speed up our model while maintaining high accuracy and conduct in-depth research on the feature fusion machine of object detection neural network.

REFERENCES

[1] 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.

[2] 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.

[3] K. Sun, B. Xiao, D. Liu, and J. Wang, ''Deep high-resolution representation learning for human pose estimation,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5693–5703.

[4] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, ''High-resolution representations for labeling pixels and regions,'' Apr. 2019, *arXiv:1904.04514*. [Online]. Available: http://arxiv.org/abs/1904.04514

[5] Y. Cao, K. Chen, C. C. Loy, and D. Lin, ''Prime sample attention in object detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11583–11591.

[6] D. G. Lowe, ''Distinctive image features from scale-invariant keypoints,'' *Int. J. Comput. Vis.*, vol. 69, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[7] H. Bay, T. Tuytelaars, and G. L. Van, *SURF: Speeded Up Robust Features*, vol. 3951. Berlin, Germany: Springer, 2006, doi: 10.1007/11744023_32.

[8] N. Dalal and B. Triggs, ''Histograms of oriented gradients for human detection,'' in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[9] J. Platt, ''Sequential minimal optimization: A fast algorithm for training support vector machines,'' Microsoft Research, Bengaluru, India, Tech. Rep. MSR-TR-98-14, Apr. 1998.

[10] S. Saha, S. A. Kamran, and A. S. Sabbir, ''Total recall: Understanding traffic signs using deep hierarchical convolutional neural networks,'' Oct. 2018, *arXiv:1808.10524*. [Online]. Available: http://arxiv.org/abs/1808.10524

[11] Y. Kageyama, A. Takano, and M. Nishida, ''Method for extracting circular road signs on the basis of scene image features,'' *IEEJ Trans. Electron., Inf. Syst.*, vol. 130, no. 10, pp. 1865–1872, 2010.

[12] Y. Kageyama, S. Asano, and M. Nishida, ''Estimation of internal area in the circular road signs from color scene image,'' *IEEJ Trans. Electron., Inf. Syst.*, vol. 124, no. 2, pp. 578–579, 2004.

[13] H. Guan, J. Li, Y. Yu, C. Wang, M. Chapman, and B. Yang, ''Using mobile laser scanning data for automated extraction of road markings,'' *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, Jan. 2014.

[14] A. R. Dubey, N. Shukla, and D. Kumar, *Detection and Classification of Road Signs Using HOG-SVM Method*, vol. 766. Singapore: Springer, 2020, doi: 10.1007/978-981-13-9683-0_6.

[15] M. Takaki and H. Fujiyoshi, ''Traffic sign recognition using SIFT features,'' *IEEJ Trans. Electron., Inf. Syst.*, vol. 129, no. 5, pp. 824–831, 2009.

[16] K. B. Lee and H. S. Shin, ''An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels,'' in *Proc. Int. Conf. Deep Learn. Mach. Learn. Emerg. Appl. (Deep-ML)*, Istanbul, Turkey, Aug. 2019, pp. 7–11, doi: 10.1109/Deep-ML.2019.00010.

[17] H.-C. Shin, K.-I. Lee, and C.-E. Lee, ''Data augmentation method of object detection for deep learning in maritime image,'' in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Busan, South Korea, Feb. 2020, pp. 463–466, doi: 10.1109/BigComp48618.2020.00-25.

[18] A. Alfarrarjeh, D. Trivedi, S. H. Kim, and C. Shahabi, ''A deep learning approach for road damage detection from smartphone images,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 5201–5204, doi: 10.1109/BigData.2018.8621899.

[19] Z. Wu, N. M. Khan, L. Gao, and L. Guan, ''Deep reinforcement learning with parameterized action space for object detection,'' in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2018, pp. 101–104, doi: 10.1109/ISM.2018.00025.

[20] X. Xiaozhu and H. Cheng, ''Object detection of armored vehicles based on deep learning in battlefield environment,'' in *Proc. 4th Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Changsha, China, Jul. 2017, pp. 1568–1570, doi: 10.1109/ICISCE.2017.327.

[21] 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.

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

[23] J. Dai, Y. Li, K. He, and J. Sun, ''R-FCN: Object detection via region-based fully convolutional networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, ''Mask R-CNN,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[25] 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

[26] 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.

[27] J. Redmon and A. Farhadi, ''YOLO9000: Better, faster, stronger,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

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

[29] W. Hai, W. Kuan, C. Yingfeng, L. Ze, and C. Long, ''Traffic sign recognition based on improved cascade convolution neural network,'' *Automot. Eng.*, vol. 42, pp. 1256–1262, Sep. 2020.

[30] Z. Cai and N. Vasconcelos, ''Cascade R-CNN: Delving into high quality object detection,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6154–6162, doi: 10.1109/CVPR.2018.00644.

[31] 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.

[32] C. Han, G. Gao, and Y. Zhang, ''Real-time small traffic sign detection with revised faster-RCNN,'' *Multimedia Tools Appl.*, vol. 78, pp. 13263–13278, Aug. 2019, doi: 10.1007/s11042-018-6428-0.

[33] K. Simonyan and A. Zisserman, ''Very deep convolutional networks for large-scale image recognition,'' Apr. 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[34] A. Shrivastava, A. Gupta, and R. Girshick, ''Training region-based object detectors with online hard example mining,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 761–769.

[35] J. Jiang, S. Bao, W. Shi, and Z. Wei, ''Improved traffic sign recognition algorithm based on YOLO V3 algorithm,'' *J. Comput. Appl.*, vol. 40, pp. 2472–2478, Apr. 2020.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, ''Generalized intersection over union: A metric and a loss for bounding box regression,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 658–666.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ''Focal loss for dense object detection,'' Aug. 2017, *arXiv:1708.02002*. [Online]. Available: http://arxiv.org/abs/1708.02002

[38] J. Zhang, J. Zhang, and S. Yu, ''Hot anchors: A heuristic anchors sampling method in RCNN-based object detection,'' *Sensors*, vol. 18, no. 10, p. 3415, Oct. 2018.

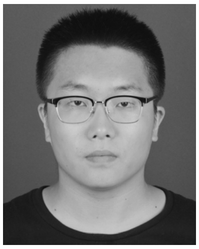
[39] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, ''Feature pyramid networks for object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[40] J. Heaton, I. Goodfellow, Y. Bengio, and A. Courville, ''Deep learning,'' *Genet. Program. Evolvable Mach.*, pp. 305–307, Mar. 2018.

[41] 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*, vol. 8693. Cham, Switzerland: Springer, 2014, doi: 10.1007/978-3-319-10602-1_48.

[42] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, ''Benchmarking robustness in object detection: Autonomous driving when winter is coming,'' Jul. 2019, *arXiv:1907.07484*. [Online]. Available: http://arxiv.org/abs/1907.07484

[43] D. Hendrycks and T. G. Dietterich, ''Benchmarking neural network robustness to common corruptions and surface variations,'' Apr. 2018, *arXiv:1807.01697*. [Online]. Available: http://arxiv.org/abs/1807.01697



JINGHAO CAO was born in Shanxi, China, in 1997. He is currently pursuing the master's degree with the School of Electronic Engineering and Optoelectronic Technology, Nanjing University of Science and Technology. His main research interests include photoelectric detection and image engineering, especially in the object detection based on deep learning algorithm. He is a member of the China Computer Federation (CCF).

JUNJU ZHANG was born in 1979. He received the Ph.D. degree in optical engineering from the Nanjing University of Science and Technology.

In recent years, he has been responsible for participated in more than 20 national, provincial or horizontal development research projects. He has published more than 30 papers in journals and conferences, of which more than 20 were indexed by SCI and EI, applied for five national invention patents, and compiled one national textbook. His main research interests include photoelectric information detection, image signal processing, photoelectric emission material theory, and preparation technology.

WEI HUANG was born in Hubei, China, in 1996. He is currently pursuing the master's degree with the School of Electronic Engineering and Optoelectronic Technology, Nanjing University of Science and Technology. His main research interests include photoelectric detection and image engineering.