Hindawi
Journal of Advanced Transportation
Volume 2022, Article ID 3825532, 16 pages
https://doi.org/10.1155/2022/3825532

Research Article

Traffic Sign Detection via Improved Sparse R-CNN for Autonomous Vehicles

Tianjiao Liang,[1,2] **Hong Bao**,[1,2] **Weiguo Pan**,[1,2] **and Feng Pan**[1,2]

[1]*Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China*
[2]*College of Robotics, Beijing Union University, Beijing, China*

Correspondence should be addressed to Weiguo Pan; ldtweiguo@buu.edu.cn

Received 9 September 2021; Revised 21 December 2021; Accepted 11 January 2022; Published 1 March 2022

Academic Editor: Zhihong Yao

Copyright © 2022 Tianjiao Liang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic sign detection is an important component of autonomous vehicles. There is still a mismatch problem between the existing detection algorithm and its practical application in real traffic scenes, which is mainly due to the detection accuracy and data acquisition. To tackle this problem, this study proposed an improved sparse R-CNN that integrates coordinate attention block with ResNeSt and builds a feature pyramid to modify the backbone, which enables the extracted features to focus on important information, and improves the detection accuracy. In order to obtain more diverse data, the augmentation method used is specifically designed for complex traffic scenarios, and we also present a traffic sign dataset in this study. For on-road autonomous vehicles, we designed two modules, self-adaption augmentation (SAA) and detection time augmentation (DTA), to improve the robustness of the detection algorithm. The evaluations on traffic sign datasets and on-road testing demonstrate the accuracy and effectiveness of the proposed method.

1. Introduction

Traffic sign detection based on computer vision plays a crucial role in the autonomous driving system. The deep neural networks have successfully applied in many fields, such as computer vision, communications [1], and networking [2]. Applying detection algorithms based on deep learning to autonomous vehicles has become a hot topic for researchers. The traffic sign detection system can automatically detect and recognize traffic signs in real traffic scenes and then transmit the results to the decision-making module to ensure that the vehicle drives safely in accordance with traffic rules.

It remains challenging to detect and recognize traffic signs in on-road scene due to their unstable features on different occasions, such as illumination, weather, and noise. These complex factors will reduce the detection accuracy of traffic signs. The datasets used to train the detection model also affect the detection accuracy due to the data collection scenes, weather conditions, and time of data collection problem.

To solve the problems mentioned above, the motivation of this study is to enable the feature extraction process by backbone more focused on the object and detect multiscales objects. In order to improve the robustness of the detection model, the dataset used for training should be as diverse as possible, which can simulate traffic signs in complex traffic scenarios.

The transformer structure [3] has recently become a hot topic due to its competitive performance especially when vision transformer (ViT) [4] and DERT [5] are proposed to make transformer applied in computer vision. The sparse R-CNN [6], which is inspired by the transformer, is a purely sparse method for object detection compared to the ordinary CNN models. It uses a series of learnable proposal boxes and features to replace the thousands of candidates generated by traditional region proposal algorithm, such as selective search [7] in R-CNN and region proposal networks (RPN) [8] in faster R-CNN. The proposed traffic sign detection method in this study is based on sparse R-CNN.

The contributions of this study are listed as follows:

(1) The proposed method multiscale sparse R (MSR)-CNN integrates coordinate attention block into the backbone network ResNeSt [9], which can improve the model to find a region of interest in images. Then, a feature pyramid network is used for multiscale detection.

(2) The data augmentation driven by complex traffic scenarios is used to make the dataset used in training more diverse.

(3) In order to improve the detection accuracy in on-road scenarios, this study designs self-adaption augmentation (SAA) in front of the MSR and detection time augmentation (DTA) module behind the MSR.

(4) This study presents an annotated traffic sign dataset called Beijing Union University Chinese Traffic Sign Detection Benchmark (BCTSDB).

The rest of this study is organized as follows: in the related work section, we introduce object detection and traffic sign detection algorithms in recent years. The details of the proposed traffic signs detection model are presented in the proposed method section. The following section focuses on its implementation and comparison with previous methods. The final section summarizes the proposed method and looks forward to the future direction.

2. Related Work

Traffic signs are usually defined as eye-catching colors in the design process to improve identifiability, so that traffic signs can be distinguished from the environmental background. Many traditional traffic sign detection methods rely on extracting features from visual information such as color, edge, and shapes.

Reference [10] proposed a traffic sign detection method based on the HOG [11] feature and SVM [12]. Firstly, the method segmented the image of traffic signs by the color threshold to remove a lot of interference and then used the maximum stable extremum region algorithm to detect the connected region. Shape is another important feature of traffic signs. Literature [13] used the shape-based method to comprehensively consider the shapes of triangle, circle, and square, and the connected component was used for shape recognition to remove the regions without traffic signs in the images.

The above detection methods are usually affected by illumination, occlusion, distortion, and scale. When applied to real traffic scenes, their slow detection speed and low accuracy cannot meet the needs of autonomous driving systems. In recent years, deep learning algorithms have been widely used in object detection tasks for their competitive performance.

The detection method based on deep learning is mainly divided into dense algorithms and dense-to-sparse algorithms [6]. The dense algorithms also called one-stage algorithms, such as the you only look once (YOLO) series [14–17], single-shot multibox detector (SSD) [18], and RetinaNet [19], which directly output the location and category of densely bounding boxes from features in a single-shot way. They directly predict anchor boxes or key points [20] densely covering spatial positions, which are built on dense candidates, and each candidate will be classified and regressed, respectively. Especially in anchor-based algorithms, for each position in the feature map (H × W), k anchor boxes need to be set, which leads to H × W × k anchors. These candidates are assigned to ground-truth object boxes in training time and then are needed non-maximum suppression (NMS) to remove redundant predictions during inference time. The dense-to-sparse algorithms are also known as two-stage algorithms. This kind of algorithm first uses region proposal algorithm (e.g., selective search [7] and RPN [8]) to select a small set of foreground regions proposals from preset dense candidates in the first stage, and then region proposals are put into the subsequent network for classification and position regression in the second stage, such as R-CNN [7], fast R-CNN [21], and faster R-CNN [8]. More researchers began to use deep learning methods for traffic sign detection. Yang et al. [22] used adversarial machine learning to generate adversarial examples in order to improve the detection robustness of autonomous vehicles but did not consider the effect of the environment on the detection. He et al. [23] presented a traffic sign detection using CapsNet [24] based on visual inspection. However, it extracts HOG feature from images, which does not contain semantic information. Dewi et al. [25] use YOLOv4 with synthetic training data to detect traffic signs. Domen et al. [26] propose an improved mask R-CNN [27] to address the full pipeline of detection with end-to-end learning, which cannot detect small and multiscale traffic signs. Cao et al. [28] improved faster R-CNN through the high-resolution backbone network [29] and prime sample strategy [30]. Xie et al. [31] proposed improved cascade R-CNN [32] for traffic-sign detection. The above methods all use a two-stage-based algorithm with high computational complexity to detect traffic signs. These methods have improved the performance of the algorithm, but the used backbone networks were designed for classification such as VGG [33] and ResNet [34], which cannot extract deeper semantic information and contextual information due to the limited receptive field size and lack of cross-channel interaction.

The sparse R-CNN method proposes a new object detection paradigm called sparse algorithms [6], which avoids RPN and replaces it with a set of N-learned object proposals. N is much smaller than the number of anchor boxes used by the dense algorithms or dense-to-sparse algorithms in the first stage. Unlike the two-stage algorithm, this method has no RPN structure and the proposal boxes are generated by a set of preset learnable parameters.

The comparisons of these methods are listed in Table 1. It can be summarized from this table that the current algorithms applied in traffic sign detection are not fully integrated with attention, multiscale, and data augmentation, which results in a decrease in the detection accuracy of the model trained on the dataset in on-road testing. In this study, our work will incorporate these three factors into the process of detecting traffic signs based on sparse R-CNN to improve the accuracy of traffic signs detection.

Table 1: Different traffic sign algorithms.

	Methods	Framework	Attention	Multiscale	Data augmentation
Traditional algorithms	Yao et al. [10]	HOG + SVM	–	–	–
	Yildiz et al. [13]	HOG + SVM	–	–	–
Dense algorithms	Yang et al. [22]	Adversarial network	–	–	✓
	He et al. [24]	HOG + CapsNet	–	–	–
	Dewi et al. [25]	YOLOv4	–	✓	✓
Dense-to-sparse algorithms	Domen et al. [26]	Mask R-CNN	√	–	–
	Cao et al. [28]	Faster R-CNN	–	✓	✓
	Xie et al. [31]	Cascade R-CNN	✓	✓	–
Sparse algorithms	Sun et al. [6]	Sparse R-CNN	–	–	–

3. Proposed Method

In order to improve the detection accuracy of traffic signs, the proposed framework is illustrated in Figure 1. It consists of two phases: training and inference. Our main contribution is the following three parts: (a) MSR including two parts: integrating coordinate attention block with backbone network ResNeSt and building a feature pyramid for multiscale detection. (b) Data augmentation for complex traffic environment. (c) The designed SAA and DTA modules are used to improve on-road detection accuracy.

3.1. Pipeline. In our proposed framework, as depicted in Figure 1, x^{tr} represents the training images, and x^{in} represents the testing images.

In the training phase, there are two modules that process images x^{tr} synchronously as follows: (1) the first module is MSR. Images x^{tr} are first augmented by the data augmentation method and then sent to the MSR to detect traffic signs. In MSR, the features are extracted using our proposed backbone network. In the feature extraction process, the acquired five-scale pyramid feature is denoted by $\{C_1, C_2, C_3, C_4, C_5\}$. The results obtained by the coordinate attention block and a 3 × 3 convolution kernel are $\{P_1, P_2, P_3, P_4, P_5\}$. Then corresponding proposal boxes and proposal features are input into the dynamic head to generate object features, finally, the loss value is calculated, and back-propagation training is carried out to obtain the final model. (2) The second module is SAA. It is a classifier that learns to divide illumination into low-, normal-, and highlight classes in the training phase.

In the inference phase, the test images are first sent to SAA trained on the training images to classify the illumination of x^{in}. According to the obtained image category from SAA, the nodes in the data augmentation channel were activated for data augmentation. These extended samples are input into the MSR to obtain detection results, and the final output results contained the target probability, category probability, and position information, which are processed by the DTA.

3.2. Detector

3.2.1. Sparse R-CNN. Sparse R-CNN avoids the manual setting of a large number of hyper parameters for candidate boxes and many-to-one label assignments. More importantly, the final prediction result can be directly output without NMS as illustrated in Figure 2. ResNeSt was used as the backbone network for feature extraction in the proposed framework.

In the proposed sparse R-CNN, we use the CIoU [35] loss for bounding box regression. CIoU solves the problem of not being able to directly optimize the parts where the bounding box and ground truth do not overlap. The distance between the two boxes, overlap rate, scale, and penalty terms are all taken into consideration, making the target box regression more stable. This can also prevent divergence in the training process. The loss function of CIoU adds an impact term $\alpha\nu$ based on the loss function of DIoU [35], which considers the length-to-width ratio between the predicted and ground-truth boxes.

The CIoU loss function is defined as follows:

$$\begin{aligned}
\mathscr{L}_{CIoU} &= 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha\nu,\\
\alpha &= \frac{\nu}{(1 - IoU) + \nu},\\
\nu &= \frac{4}{\pi^2}\left(\left(\arctan\frac{w^{gt}}{h^{gt}}\right) - \arctan\frac{w}{h}\right)^2,
\end{aligned} \tag{1}$$

where α is a trade-off parameter, and ν is a parameter used to measure the consistency of the aspect ratio. Furthermore, $\rho(.)$ is the distance between the central points of the two boxes, and c is the diagonal length of the smallest enclosing box covering the two boxes.

Starting from a sparse set of learnable proposals, a sparse R-CNN generates proposal boxes to extract the region of interest (ROI) and proposal features to learn ROI features. Both are learnable parameters. The dimensions of a learnable proposal box are $N \times 4$, where N represents the number of object candidates, generally ranging from 100 to 300, and there are four boundaries of the object box. The network sets a fixed number of boxes as learning parameters. The dimension of the learnable proposal feature is $N \times d$, where d represents the dimension of a feature, which is generally 256. The ROI feature extracted by the proposal boxes generates a one-to-one interaction to supplement high-level feature information such that the features of the ROI are more conducive to location and classification. The interaction design is called a dynamic instance interactive head. It binds

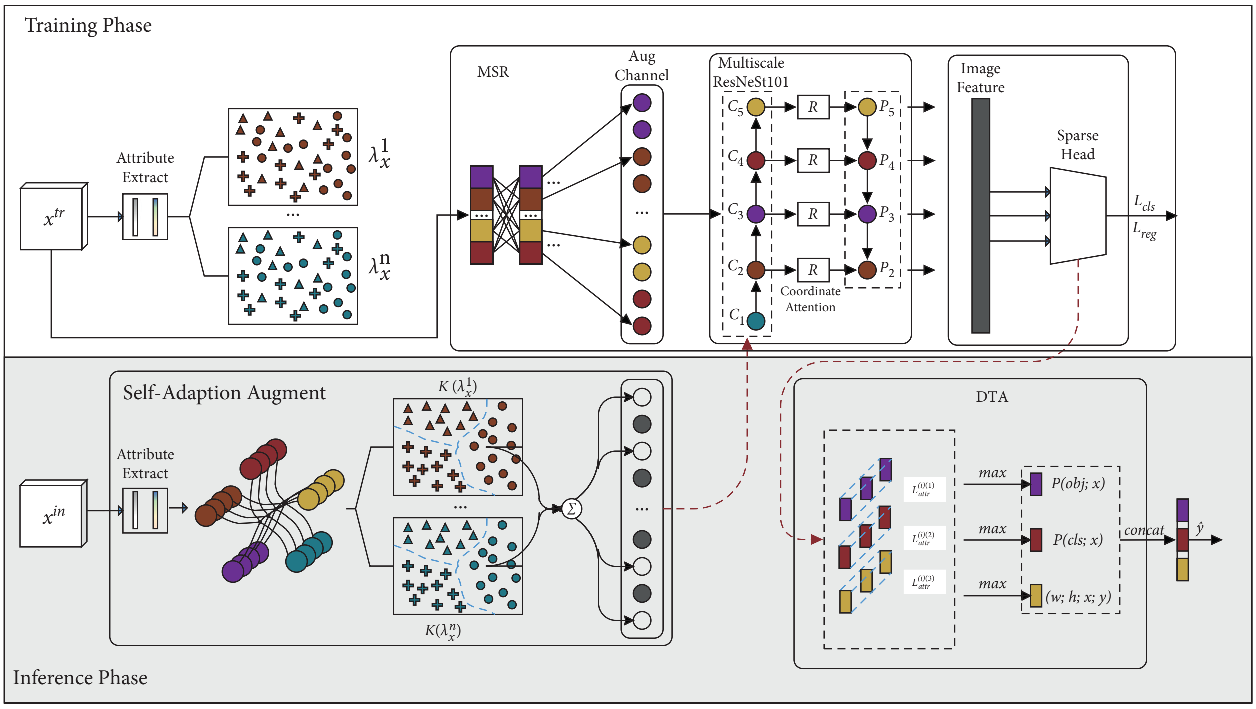

Figure 1: Proposed detection framework.

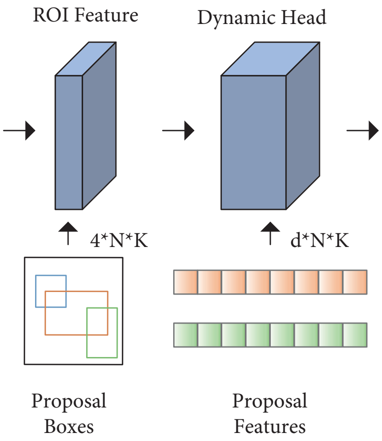

Figure 2: Sparse head.

the proposal box, ROI, and feature vector and detects each ROI separately without an NMS operation. Thus, the candidate boxes can be sparse, and the interactions between features can also be sparse. The backbone network extracts a feature map, and each proposal box and proposal feature are fed into its exclusive dynamic head to generate the object feature.

The matching cost is defined as follows:

$$\mathscr{L}_{attr} = \lambda_{\text{cls}} \cdot \mathscr{L}_{\text{cls}} + \lambda_{L1} \cdot \mathscr{L}_{L1} + \lambda_{ciou} \cdot \mathscr{L}_{ciou},$$
$$\mathscr{L}_{\text{joint}} = \sum_{i=1}^{m} \alpha^i \mathscr{L}_{attr}^i. \tag{2}$$

Here, $\mathscr{L}_{cls}$ and $\mathscr{L}_{L1}$ are the focal loss and *L1* loss, respectively. Moreover, $\mathscr{L}_{ciou}$ represents the CIoU loss, and λ_{cls}, λ_{L1}, and λ_{ciou} are the coefficients of each component. The final loss $\mathscr{L}_{\text{joint}}$ is the sum of all pairs normalized by the number of objects inside the training batch.

Sparse R-CNN can be seen as a new detection paradigm that has changed the framework of the dense detector and the dense-to-sparse detector by abandoning the concepts of anchor boxes or reference points.

3.2.2. Backbone Network. Convolutional neural networks are originally designed for image classification. Although they have competitive performance in the classification task for the limited receptive field size and lack of cross-channel interaction, these networks will be limited in the field of object detection and image segmentation. Object detection networks with cross-channel representations can solve these problems. Reference [9] proposed a ResNeSt-based split-attention blocks. Compared to the ResNet, it does not require additional calculations. ResNeSt draws on the idea of the ResNeXt network [36], dividing the input into k pieces, each marked as cardinal 1−k, and then splitting each cardinal into R pieces, marked as split 1−r; hence, there are $G = k \times R$ pieces in the total group. The structure of ResNeSt is shown in Figure 3.

In the proposed method, average pooling with a kernel size of 3 × 3 was used to reduce the spatial dimensions, and the 7 × 7 convolution was replaced by three 3 × 3 convolutions, which ensured that the receptive field remained the same and reduced the number of parameters. A 2 × 2 average pooling is added before the 1 × 1 convolution with a step size of two in the jump connection.

We construct a pyramid with a five-scale feature map $\{C_1, C_2, C_3, C_4, C_5\}$, where the subscript indicates the

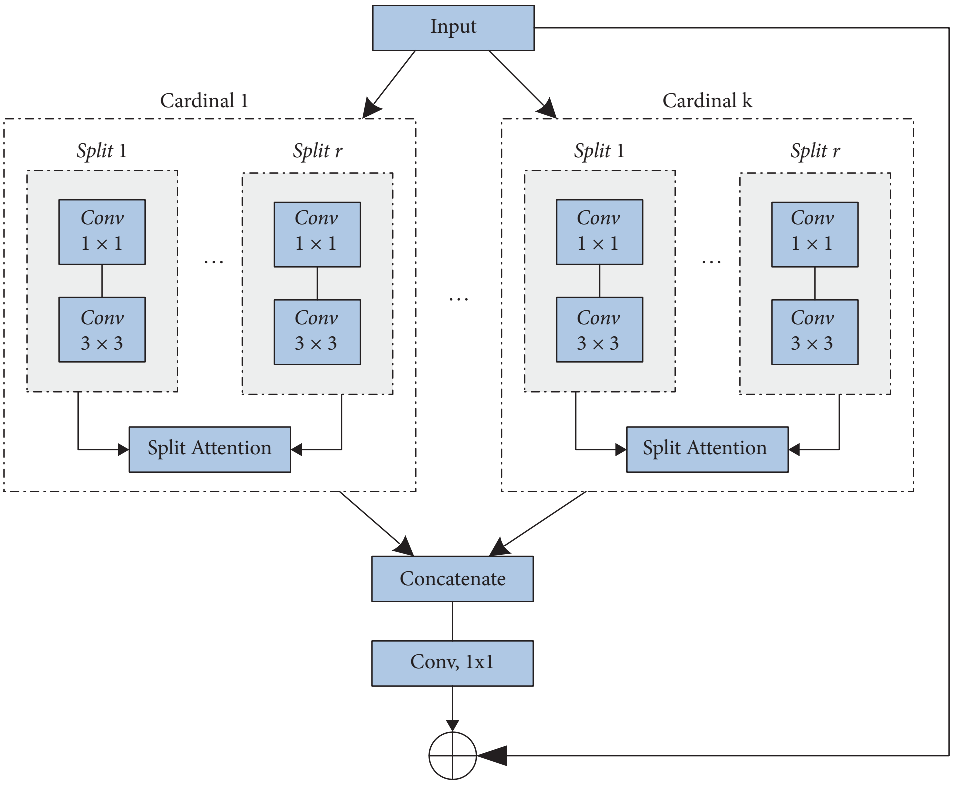

Figure 3: ResNeSt block.

pyramid level. Each time, the level is increased by one, and the resolution size is reduced by half. The feature maps were extracted by top-down convolution to reduce the degradation that occurred as the depth of the convolutional layers increased, and all maps had 256 channels. The five features at different scales are processed by the multiscale coordinate attention block to enhance attention to the traffic signs. Then, the outputs are processed by a 3 × 3 convolution to obtain feature maps $\{P_1, P_2, P_3, P_4, P_5\}$, which contain both high-resolution spatial information and low-resolution semantic information. Figure 4 illustrates the structure of the multiscale attention backbone network.

3.2.3. Multiscale Coordinate Attention. Existing attention methods in computer vision, such as SENet [37], BAM [38], and CBAM [39], only consider local area information or do not consider spatial information. Therefore, in the proposed method, we used the multiscale coordinate attention method [40]. It uses two one-dimensional global pooling operations to aggregate the input features along the vertical and horizontal directions into two separate direction-aware feature maps. Then, the two feature maps with the embedded specific direction information are coded into two attention maps, each of which captures the long-distance dependency of the input feature map along that spatial direction. The location information can be saved in the generated attention map. Then, the two attention maps are applied to the input feature maps by multiplication to emphasize the representation of the attention region.

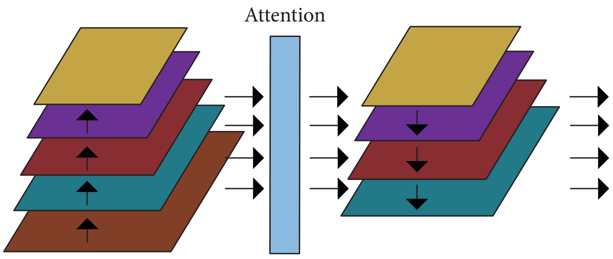

Figure 4: Backbone network.

The specific structure of the coordinate attention block is illustrated in Figure 5. In the proposed method, we use the coordinate attention block to extract the pyramid features and obtain the traffic sign features of different scales. As shown in Figure 6, the first column is the original image, the second column is the feature map without the attention mechanism, and the third column is the feature map with the attention mechanism. Using the attention mechanism can help the network to find the region of interest in images.

3.3. Data Augmentation. Deep convolution neural networks have been successfully applied in the field of computer vision. This type of method is data driven and requires a large amount of training data. As the network architecture becomes deeper, more parameters need to be learned. More data are required to allow the model performance to become superior. In a complicated traffic environment, especially in China, many factors affect traffic sign detection, such as

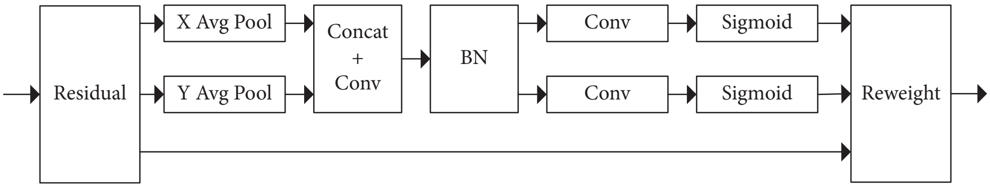

FIGURE 5: Coordinate attention block.

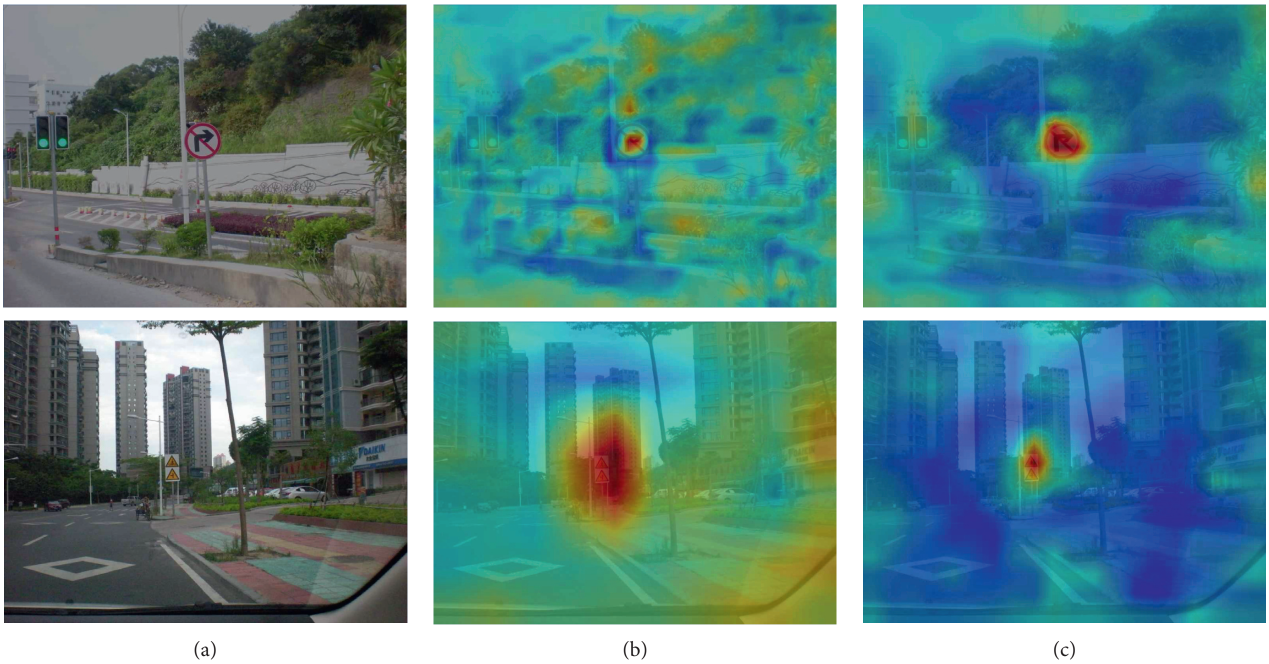

FIGURE 6: Visualization of feature maps. (a) The original image, (b) w/o. attention and (c) w/. attention.

illumination, weather, and noise. However, traffic sign datasets are not diverse enough, which do not contain data from different seasons, different times, and different weather, and only include traffic signs under certain conditions. This will affect the detection effect of the system in the actual environment. In the proposed method, we use data augmentation to improve the robustness of the model.

Dan and Dieterich [41] used data augmentation methods such as adding noise and blur in their proposed framework, but these methods are not suitable for the environment of autonomous driving. Therefore, the data augmentation method used in this study mainly simulates the complex environment of autonomous driving to make the detection model more stable.

Twenty data augmentation methods were used in the proposed method, as shown in Figure 7. They can be divided into two categories: pixel-level and spatial-level methods. Pixel-level methods change the input image, leaving other properties such as bounding boxes and the spatial position of the object unchanged. The spatial-level method changes both the spatial information and object position of the input image. The main purpose of this kind of augmentation method is to simulate the interference factors in complex traffic scenarios.

We also design a box-level data augmentation to supplement the traffic sign data that appear less in the dataset. It replaces the objects in the bounding box and blurs the border. It uses transform T to mix the two images I_b and I_o to create a new image I_a.

$$\begin{aligned} T &= I_o \times M + I_b \times (1 - M), \\ I_a &= \alpha(T(x, y)). \end{aligned} \tag{3}$$

In the formula, I_o is the object image, I_b is the background image, and M is a binary mask of object using ground-truth annotations. I_b and I_o are selected images from datasets. The proposed method extracts the object region from I_o and proportions it to the object region in I_b. This result in a gradient at the junction, and the method further uses Gaussian kernel α to blur this junction and alleviate abnormal fitting caused by drastic gradient changes. The box-level augmentation and results are shown in Figure 8 and 9.

3.4. Inference Phase

3.4.1. Self-Adaption Augmentation (SAA). In order to reduce the influence of illumination factors in the on-road detection stage, we proposed the SAA module.

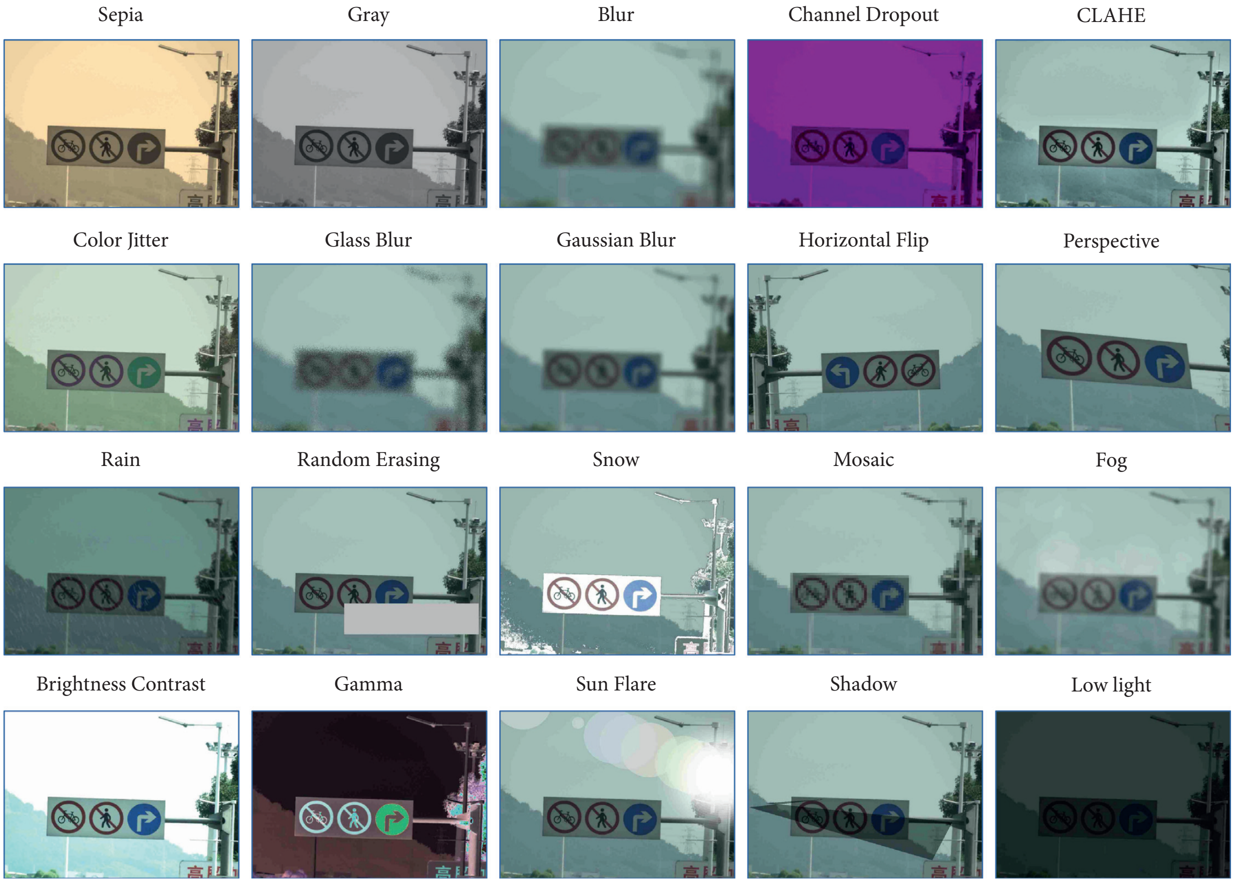

Figure 7: Examples of data augmentation.

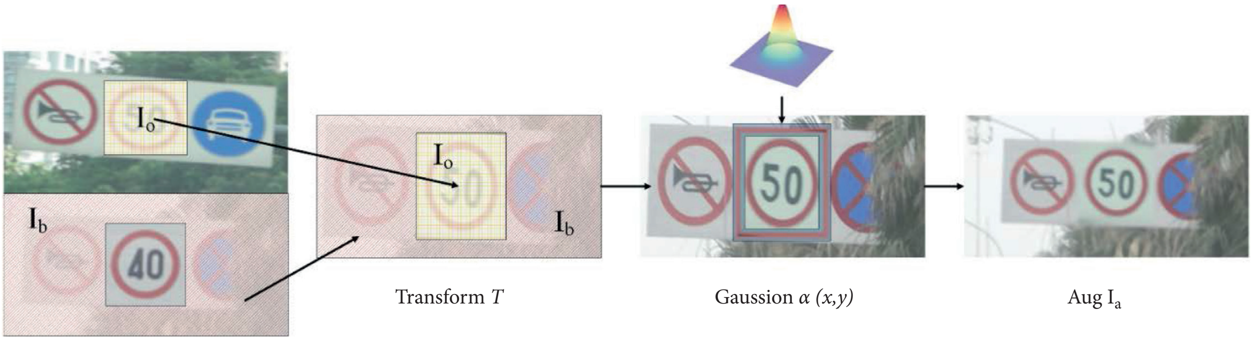

Figure 8: Box-level data augmentation.

The illumination is the greatest influence factor on the processing of traffic sign detection. In the proposed framework, a VGG-16 neural network was trained to classify the illumination of the input image. The illumination is divided into low-, normal-, and high-light classes. In the on-road test stage, when the image is input to the SAA module, it classifies the input image according to the illumination. If the image is under low- or high-illumination conditions, the brightness of the image is adjusted and then processed by the DTA module. We adjusted the original image to two different degrees according to the light intensity (Figure 10). Figure 10(a) is the original image, whereas Figure 10(b) and 10(c) shows the results of different adjustments of the original image. The specific process is shown in Figure 11, where three enhanced sample images are generated by SAA. Then, the augmented samples are input to the detector for processing. Specifically, if the image is under normal illumination, the trained detection model will be directly used for detection.

3.4.2. Detection Time Augmentation (DTA). The trained detection model used in real traffic scenarios may output false or missed detections, which will cause autonomous vehicles to make wrong decisions and cause traffic accidents. To increase the robustness of the model, we propose a DTA

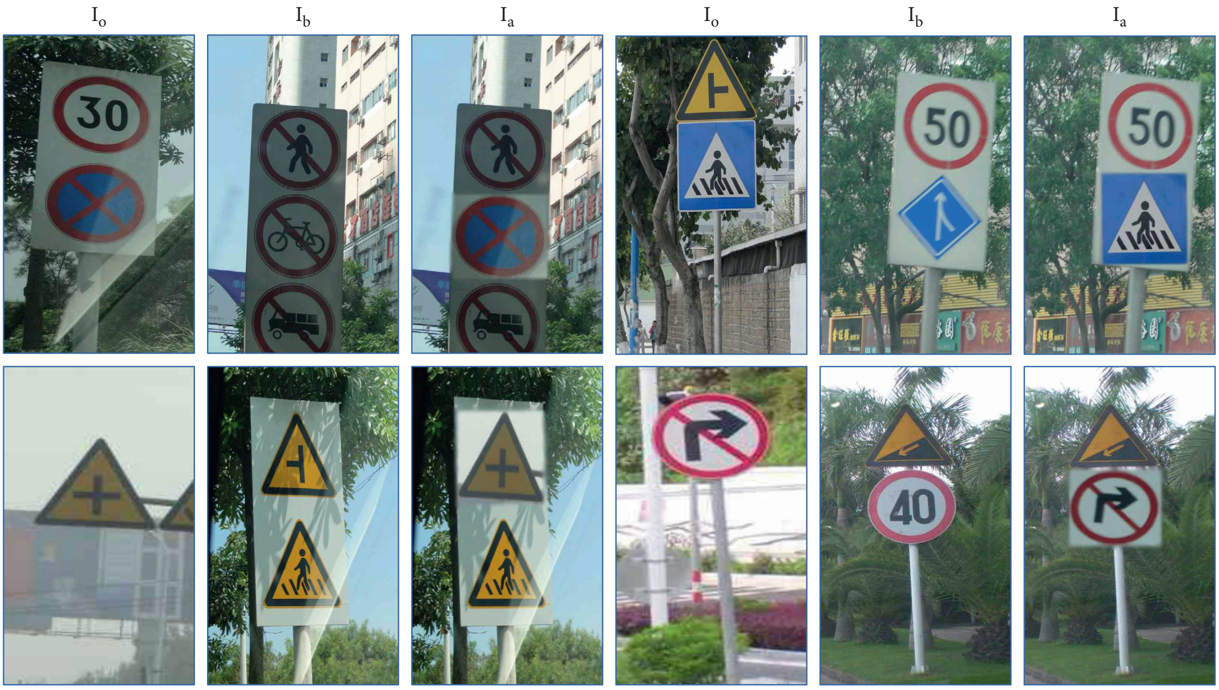

Figure 9: Results of box-level data augmentation.

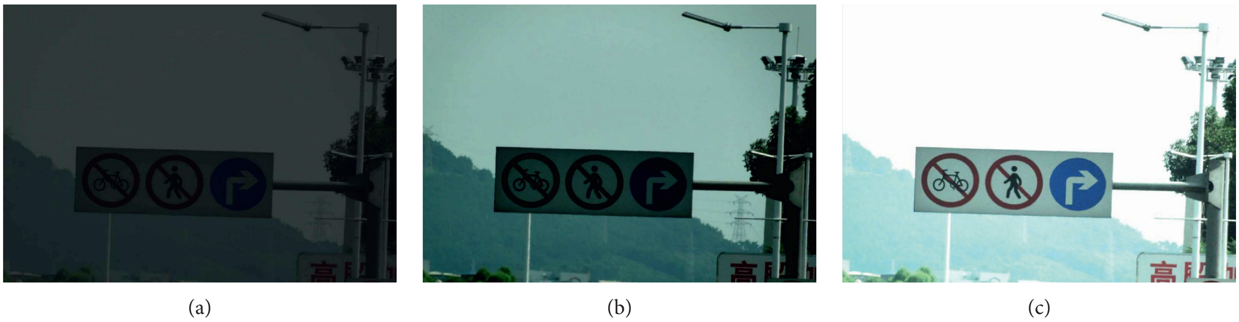

Figure 10: Illumination of augmentation results.

method for traffic sign detection, as illustrated in Figure 12. It first applies SAA to the input image to generate multiple samples. Then, the augmented images are processed by the trained model, and different results are obtained, which can be divided into three parts: the probability that the image contains the object $P(obj, x)$, the category probability of the object $P(cls, x)$, and the location information of the bounding boxes (w, h, x, y).

The framework proposed in this study uses a voting mechanism for the output results, determines whether the input image contains the detection object, and outputs the result:

$$\mathrm{Obj} = \begin{cases} 1, & \sum_{i=1}^{A} \mathrm{obj}\left(X^{i}\right) \geq \lfloor \frac{A}{2} \rfloor, \\ 0, & \sum_{i=1}^{A} \mathrm{obj}\left(X^{i}\right) < \lfloor \frac{A}{2} \rfloor. \end{cases} \tag{4}$$

Here, X^i is the *i-th* augmented image, and A indicates the augmentation methods used in the proposed framework.

If the augmented image contains the object, $\mathrm{obj}(\cdot)$ is 1; otherwise, it is 0. The final Obj value depends on the statistical results of each augmented sample image, as follows:

$$\mathrm{Cls} = \underset{c\in C}{\mathrm{Argmax}}\left(f(c)\right),\ \mathrm{cls} \geq \mathrm{threshold}. \tag{5}$$

The object category Cls is obtained according to the above equation, where C is the total category trained in the detection model, and $f(c)$ is used to count the categories of detected objects in the augmented image based on whether cls is larger than the preset threshold. The final bounding box coordinates are calculated from the average coordinates of the same detection object on different images, as shown in the following equation. Finally, L_j represents the coordinate position of the same object on different images:

$$Bbox = \mathrm{Avg}\left(\sum_{j=1}^{A} L_j\right). \tag{6}$$

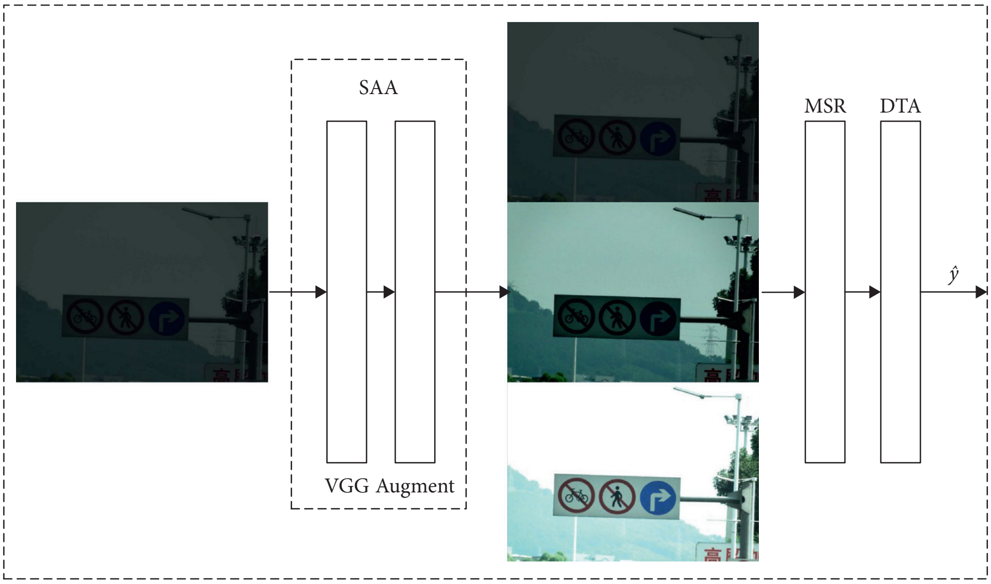

FIGURE 11: Self-adaption preprocessing.

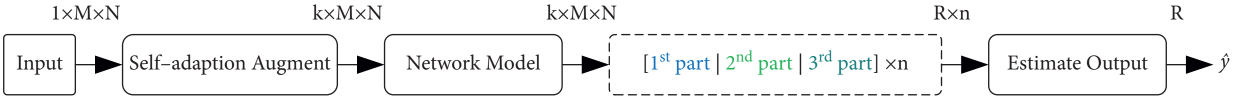

FIGURE 12: DTA.

4. Datasets

Traffic sign detection is an important task in the field of autonomous driving. Although some general image datasets, such as VOC, ImageNet, and MSCOCO, contain images of traffic signs, these data cannot be used to train traffic sign detection models. The accuracy and robustness of a network trained with different traffic sign datasets in an actual environment are different. The widely used German Traffic Sign Detection Benchmark (GTSDB) [42] contains three types of traffic signs (mandatory signs, warning signs, and prohibitory signs) and consists of 900 images with a size of 1360 × 800. The dataset published by the Laboratory for Intelligent and Safe Automobiles (LISA) [43] includes 47 types of US traffic signs, with 7,855 images and 6,610 signs ranging in size from 6 × 6 to 167 × 168 pixels taken from American traffic video frames. The traffic signs contained in these datasets are quite different from Chinese traffic signs in color and shape; hence, models trained with datasets such as GTSDB or Lisa cannot be directly applied to the recognition of Chinese traffic signs.

Tsinghua–Tencent 100K (TT-100K) [44] is a public dataset collected in China, containing 16,000 images and consisting of 27,000 traffic sign instances divided into 211 categories. The Chinese Traffic Sign Dataset (CTSD) published by the Chinese Academy of Sciences contains 1,100 images divided into 700 training images and 400 testing images with sizes of 1024 × 768 and 1280 × 720. The CSUST Chinese Traffic Sign Detection Benchmark (CCTSDB) [45] expands the CTSD by adding 5,200 images collected from the highway with a size of 1000 × 350. Models trained on these public datasets cannot be applied to real traffic scenarios.

In this study, we present a collected and annotated dataset of traffic signs named Beijing Union University Chinese Traffic Sign Detection Benchmark (BCTSDB). The autonomous vehicle used to collect data equipped with sensors is shown in Figure 13. The sensing system consists of a millimeter-wave radar, monocular camera, lidar, infrared camera, GPS receiver, and ultrasonic radar. The monocular camera is used to record traffic scenes video.

This dataset includes 15,690 images and 25,243 annotations with image sizes of 1024 × 768, 1280 × 720, 1000 × 350, and 912 × 684. Figure 14 shows the sample images from this dataset. The label categories are prohibitory, mandatory, and warning, with 12,705, 8,193, and 4,345 instances in the dataset, respectively. These data were collected from fifteen cities of China: Beijing, Changshu, Nantong, Yiwu, Tianjin, Shenzhen, Baoding, Shijiazhuang, Yan, Anyang, Zhengzhou, Kaifeng, Jingzhou, Shanghai, and Fuzhou. The image data were collected for different scene types such as urban streets, highways, and viaducts. The dataset is available at https://github.com/ltjcherry/BCTSDB.

5. Experiments

5.1. Experimental Setup. The experimental parameters of the proposed model are summarized in this section. The computer configuration included two NVIDIA TITAN V graphics cards, with a total of 24 GB VRAM. Pytorch was used to implement the network structure. Adam was used as an optimizer with a weight decay [46] of 0.0001. The initial

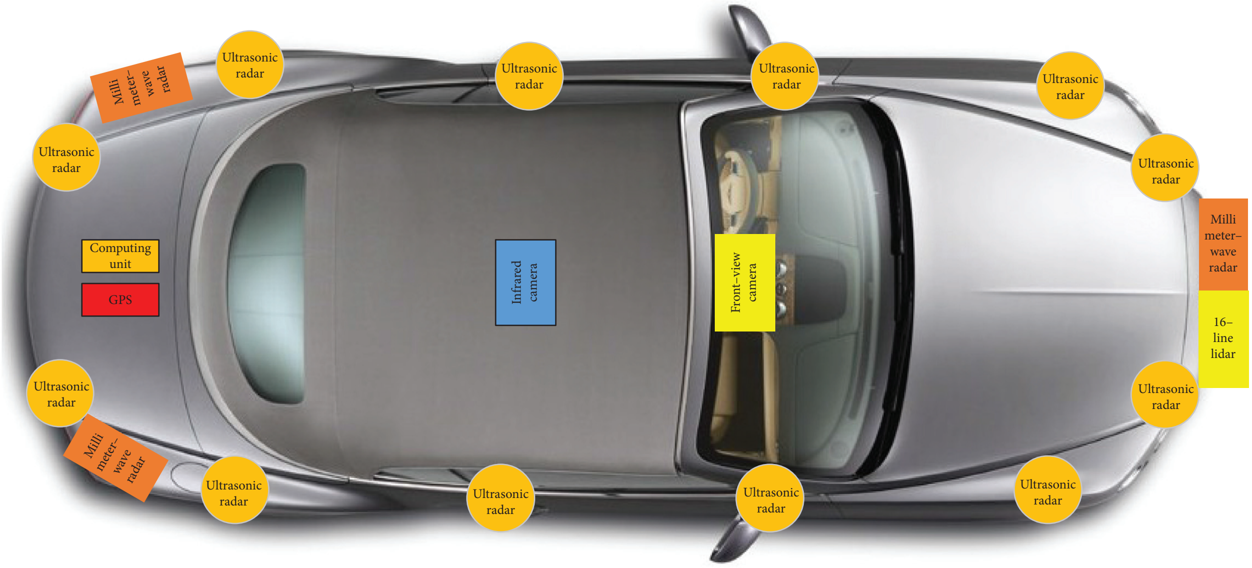

FIGURE 13: Autonomous vehicle sensors' layout.

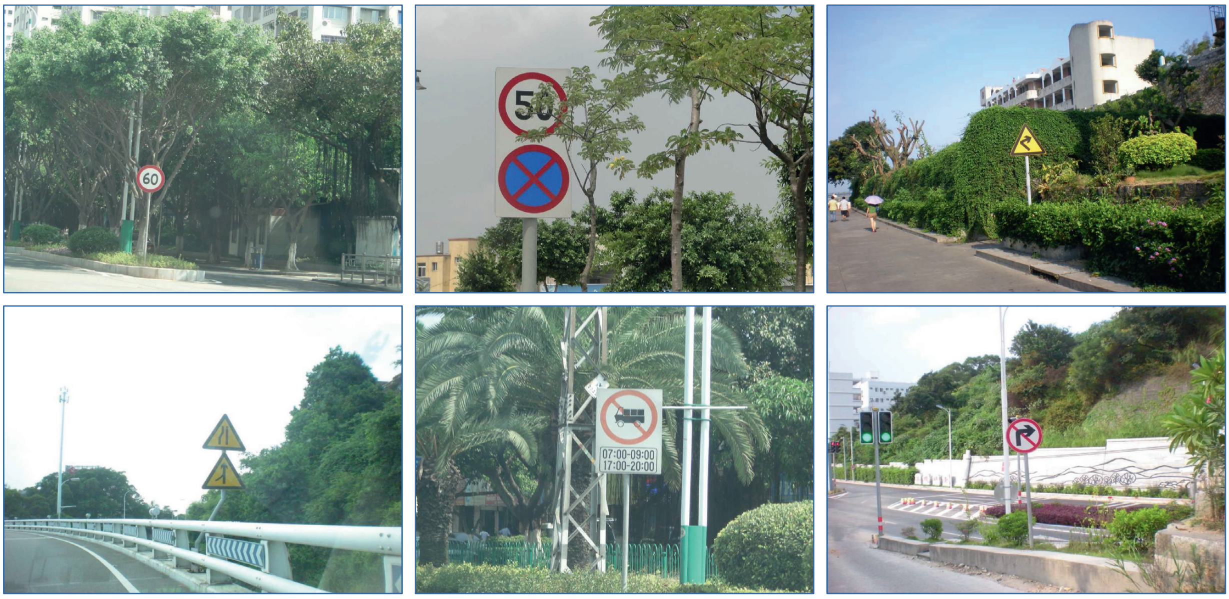

FIGURE 14: Example images of BCTSDB dataset.

learning rate was set to 2.5×10^{-5}. The backbone network was initialized using pretrained weights from ImageNet [47] and Xavier [48] for new layers. The default number of proposal boxes, proposal features, iterations, and SAA were 100, 100, 6, and 3, respectively. Our method was evaluated on the BCTSDB and TT-100K. We replace BN with SynBN to accelerate model training, and the training parameters are no longer affected by the number of GPUs, which has been successfully used in MegDet [49].

5.2. Performances on BCTSDB. The experiment used average precision (AP) to compare different models and their accuracies. Both recall and precision are considered in the calculation of the AP, which takes the average value of the precision rate at each recall point from 0 to 1. Precision is the ratio at which the original object is accurately detected, and recall is the proportion of labeled objects in the image that are detected correctly.

We randomly divided the BCTSDB into 14,591 training set images containing 23,440 annotated labels and 1,099 test set images containing 1,803 annotated labels. Figure 15 shows the detection results of BCTSDB. The top part of the figure is the original image, and the bottom part is the detection result, which displays the detected bounding box on the images. It can be clearly seen in Figure 15 that the method proposed in this study can effectively detect traffic signs.

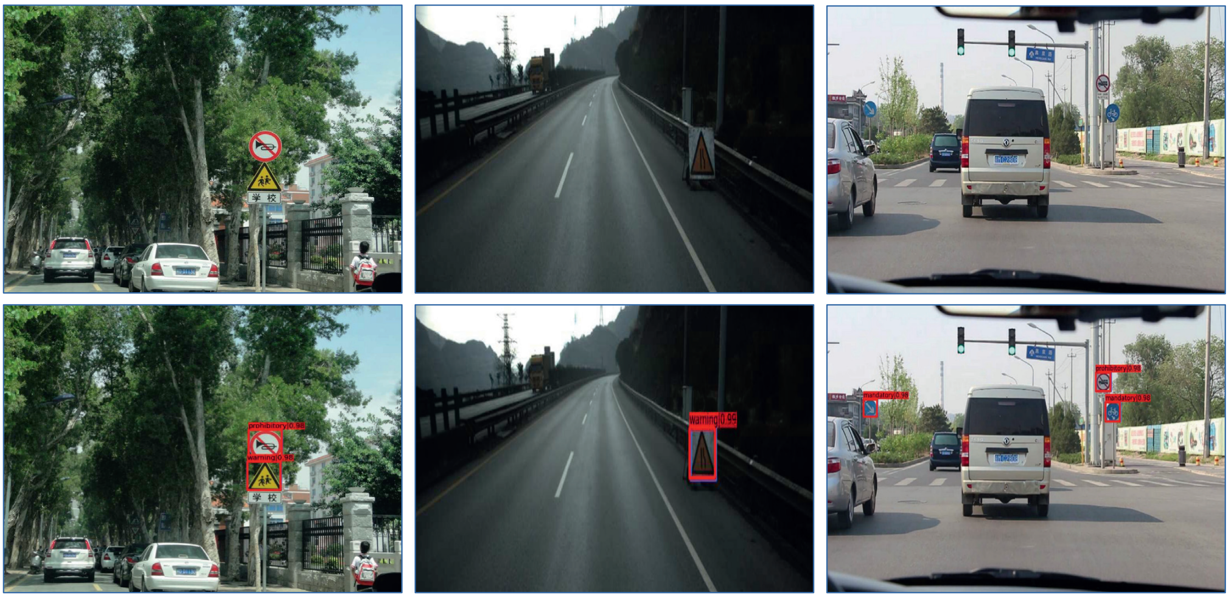

Figure 15: Traffic sign detection results on BCTSDB dataset.

The loss curves of the different methods with respect to the training epoch are shown in Figure 16. It also shows that our proposed model converges faster and has a lower loss value.

The experiments in this study proved that DTA can effectively improve the accuracy and robustness of the model. For instance, in Figure 17, the upper image shows that the network has missed detections without the DTA module, and the lower image shows the detection result after using DTA, revealing that our method can detect all traffic signs in the image.

To verify the effectiveness of each proposed module, we conducted ablation experiments. To the MSR baseline, data augmentation, multiscale attention, DTA, and SynBN were gradually added. The same parameters and training schemes were used for each ablation experiment. The result of ablation studies for the BCTSDB is listed in Table 2. The AP50 and AP75 of our proposed method obtain 3.3% and 3.7% improvement, respectively, based on the sparse R-CNN with ResNeSt101.

We further evaluate the effectiveness of commonly used data augmentation and box-level data augmentation, as listed in Table 3. Experiments have proved that both the commonly used data augmentation methods and box-level augmentation can improve the detection accuracy of the model.

Comparisons among the different methods are presented in Table 4, which lists the detection results based on RetinaNet, YOLOv3, YOLOv5, faster R-CNN, cascade R-CNN, and sparse R-CNN with different backbones. Our method can obtain more competitive results, with AP50 and AP75 values of 99.1% and 96.2%, respectively, which are better than the results of other methods. Compared to the original sparse R-CNN with ResNet101, our model can improve AP50 and AP75 by 4.4% and 12%, respectively. It can also be seen that the method proposed in this study improves the detection accuracy and has little impact on FPS.

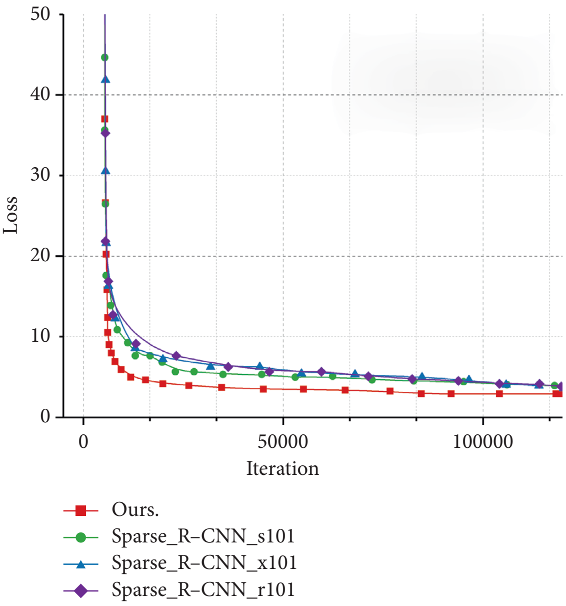

Figure 16: The loss curves of different methods.

5.3. *Performances on TT-100K.* We also evaluated the method proposed in this article on the TT-100K data, using 6,103 images containing 16,524 labeled boxes as the training dataset and 3,067 images containing 8,181 labeled boxes and 221 categories of traffic signs as the test dataset. It can be seen from the comparison results as Table 5 lists that the detection accuracy of the proposed method is greatly improved compared with the existing algorithms. Compared to the original sparse R-CNN with ResNet101, our model can

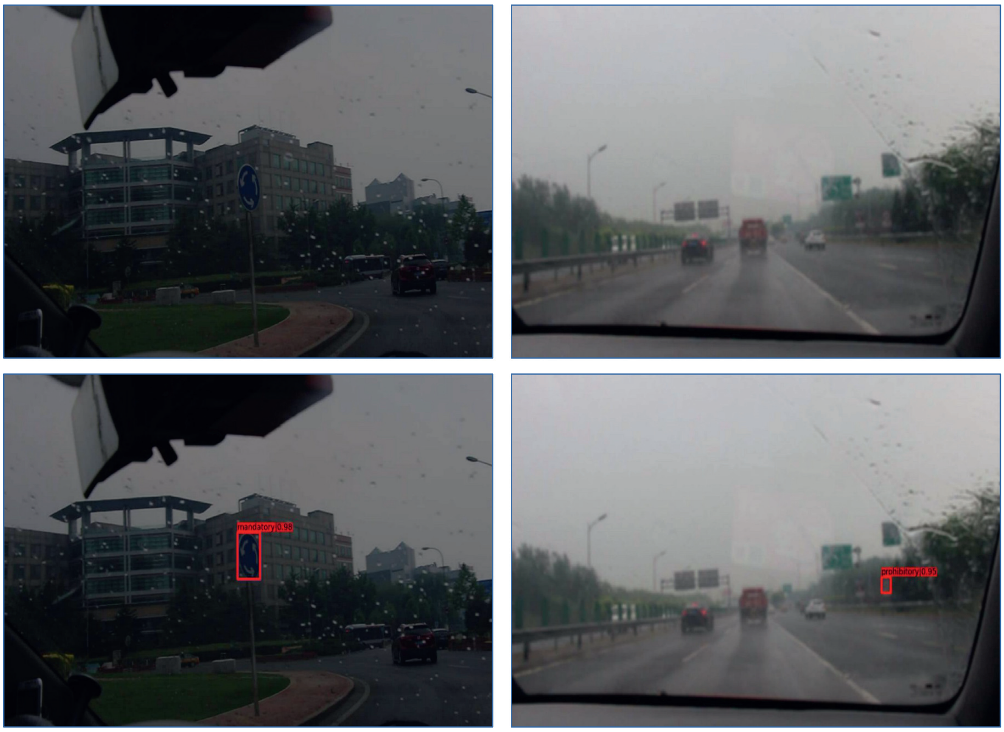

Figure 17: Traffic sign detection results with DTA.

Table 2: Ablation studies on each component in our method.

Data augmentation	SynBN	Attention	DTA	AP50	AP75
				95.8	92.5
✓				98.1	94.2
✓	✓			98.4	94.8
✓	✓	✓		98.7	95.3
✓	✓	✓	✓	99.1	96.2

Table 3: Comparisons with data augmentation.

Method	Backbone	Augmentation	Box-level augmentation	AP	AP50
RetinaNet	ResNet101	–	–	61.3	92.7
		Y	–	63.8	94.8
		Y	Y	64.1	95.2
Faster R-CNN	ResNet101	–	–	70.2	94.7
		Y	–	72.4	96.1
		Y	Y	72.7	96.5

improve AP50 and AP75 by 7.9% and 8.9%, respectively and run at 18 fps using our proposed backbone network. The TT-100K dataset itself has the problem of an uneven distribution of image categories, and hence the current detection algorithm results are generally low for TT-100K. The bottom row of Figure 18 illustrates the detection results of the proposed method on TT-100K.

5.4. On-Road Testing. To further evaluate the model performance in real traffic scenarios, we assemble the model into the autonomous vehicles. The autonomous vehicles used for the on-road testing are shown in Figure 19. And the on-road test area is illustrated in Figure 20. This area is the urban road environment competition in China's Intelligent Vehicle Future Challenge. The area includes many types of intersections, urban road traffic signs, and road markings.

In this part, we used the maximum detectable distance to evaluate our proposed method and calculated the average distance with the standard deviation as the error according to the images collected during the autonomous driving mode. In Figure 21, the box represents the quartiles, the line inside the box represents the median of the distance, and the ends of the boxes represent the minimum and maximum of each set of distances. It can be concluded from the experimental results that our method can detect traffic signs up to

Table 4: Detection result on BCTSDB.

Method	Backbone	AP	AP50	AP75	FPS
RetinaNet	ResNet50	59.7	89.4	71.2	24
RetinaNet	ResNet101	61.3	92.7	73.5	18
YOLOv3	Darknet53	59.5	92.7	70.4	33
YOLOv5	CSPDarknet	72.3	97.3	90.3	45
Faster R-CNN	ResNet101	70.2	94.7	86.0	18
Cascade R-CNN	ResNet101	75.8	96.7	92.5	13
Cascade R-CNN	ResNeXt101	77.2	97.3	93.8	13
Cascade R-CNN	ResNeSt101	77.8	97.5	94.1	10
Sparse R-CNN	ResNet50	67.6	94.2	83.8	25
Sparse R-CNN	ResNet101	69.8	94.7	84.2	22
Sparse R-CNN	ResNeXt101	73.4	95.1	91.4	23
Sparse R-CNN	ResNeSt101	76.5	95.8	92.5	20
Ours	Multiscale ResNeSt101	**78.9**	**99.1**	**96.2**	18

The meaning of the bold values is the accuracy of the proposed detection algorithm (%).

Table 5: Detection result on TT-100K.

Method	Backbone	AP	AP50	AP75	FPS
RetinaNet	ResNet101	17.5	32.3	16.9	18
YOLOv3	Darknet53	16.1	30.5	13.2	33
YOLOv5	CSPDarknet	26.7	34.8	28.6	45
Faster R-CNN	ResNet101	39.8	50.0	47.6	18
Cascade R-CNN	ResNet50	23.5	29.7	27.9	17
Cascade R-CNN	ResNet101	28.2	34.7	33.0	13
Cascade R-CNN	ResNeXt101	27.4	34.4	32.2	13
Cascade R-CNN	ResNeSt101	30.9	37.0	36.2	10
Sparse R-CNN	ResNet101	33.9	45.2	39.8	22
Sparse R-CNN	ResNeSt101	38.6	50.0	45.4	20
Ours	Multiscale ResNeSt101	**42.2**	**53.1**	**48.7**	18

The meaning of the bold values is the accuracy of the proposed detection algorithm (%).

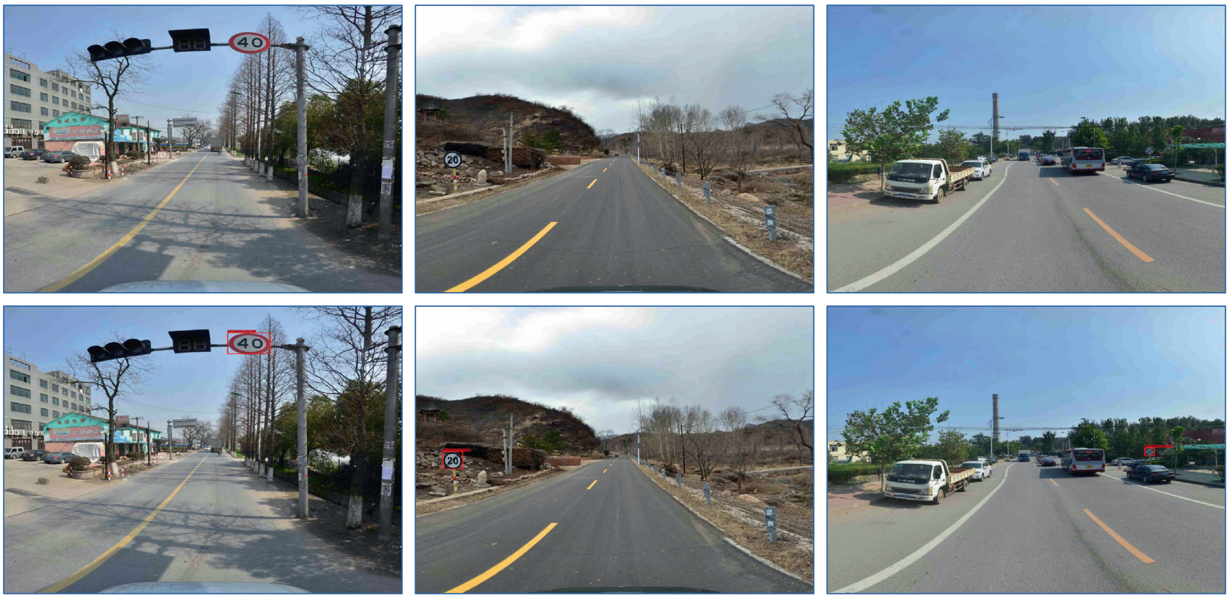

Figure 18: Traffic sign detection results on TT-100K dataset.

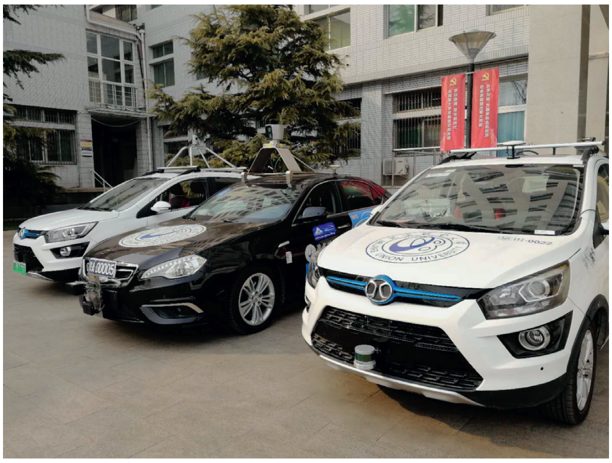

Figure 19: Autonomous vehicles for on-road testing.

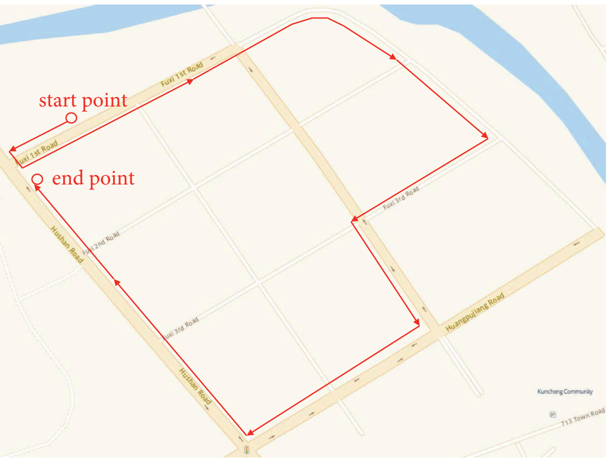

Figure 20: On-road test area.

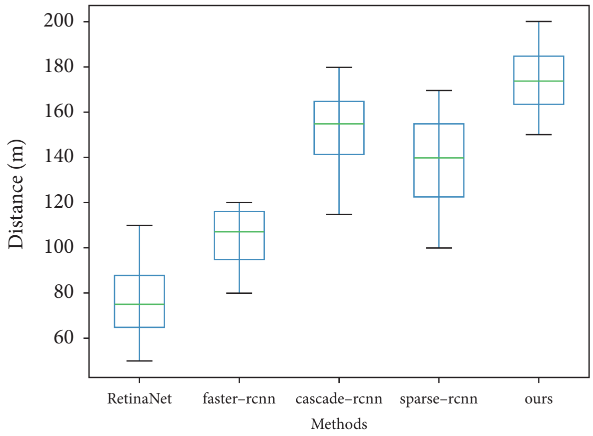

Figure 21: Detection distance.

200 meters away, which provides more processing time for the decision-making module and control module of the autonomous driving system.

6. Conclusion

Traffic sign detection can achieve high accuracy in an ideal environment, but when applied to autonomous vehicles, the detection accuracy will be reduced due to complex traffic scenes. In this study, we contribute to this gap through an improved sparse R-CNN method. The main contribution of this study is to integrate the attention mechanism and feature pyramid into the backbone network, so that the extracted features can focus on useful information. The data augmentation method is used to simulate complex traffic scenes. We also present a traffic sign dataset BCTSDB. The use of SAA and DTA modules can make the on-road traffic sign detection of the autonomous vehicle more robust. The experimental results on the BCTSDB and TT-100K datasets verify the effectiveness of the method in this study. The AP50 and AP75 of proposed method are 99.1% and 96.2% for BCTSDB, and 53.1% and 48.7% on TT-100K, respectively, which indicates that our proposed method achieves state-of-the-art results.

In the future, our work will focus on how to improve the high accuracy detection algorithm to achieve fast detection speed. The XAI [50] development may provide a quick solution to this problem. While consider applying HD map, V2X and 5G technologies to autonomous driving are a way to accelerate the industrialization.

ABBREVIATIONS

AM:	Attention module
AP:	Average precision
BCTSDB:	BUU Chinese Traffic Sign Detection Benchmark
CIoU:	Complete IoU
CTSD:	Chinese Traffic Sign Dataset
DIoU:	Distance-IoU
DTA:	Detection time augmentation
GIoU:	Generalized IoU
GTSDB:	German Traffic Sign Detection Benchmark
HOG:	Histogram of oriented gradients
IoU:	Intersection over union
Lisa:	Laboratory for Intelligent and Safe Automobiles
MSR:	Multiscale sparse R-CNN
NMS:	Nonmaximum suppression
ROI:	Region of interest
SAA:	Self-adaption augmentation
SIFT:	Scale-invariant feature transform
SSD:	Single-shot multibox detector
SVM:	Support vector machine
SynBN:	Synchronized BN
TT-100K:	Tsinghua–Tencent 100K
YOLO:	You Only Look Once.

Data Availability

The image data used to support the findings of this study are available from the corresponding author upon request.

Consent

Not applicable.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

T.L. and W.P. conceptualized the study; H.B. was responsible for methodology; T.L. and W.P. provided software; F.P. validated the study; T.L. prepared the original draft; T.L. and W.P. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant nos. 61802019, 61932012, and 61871039, in part by the Beijing Municipal Education Commission Science and Technology Program under grant nos. KM201911417003, KM201911417009, and KM201911417001, and in part by the Beijing Union University Research and Innovation Projects for Postgraduates under grant nos. YZ2020K001 and YZ2021K001.

References

[1] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[3] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, Long Beach, California, USA, May 2017.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: transformers for image recognition at scale," 2021, http://arxiv.org/abs/2010.11929.

[5] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End object detection with transformers," in *Proceedings of the Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., pp. 213–229, Springer International Publishing, Glasgow, UK, August 2020.

[6] P. Sun, R. Zhang, Y. Jiang et al., "Sparse R-CNN: end-to-end object detection with learnable proposals," in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Article ID 14458, Nashville, TN, USA, June 2021.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[9] H. Zhang, C. Wu, Z. Zhang et al., "Resnest: split-attention networks," 2020, http://arxiv.org/abs/2004.08955.

[10] C. Yao, F. Wu, J. Chen, L. Hao, and Y. Shen, "Traffic sign recognition using HOG-SVM and grid search," in *Proceedings of the 2014 12th International Conference on Signal Processing (ICSP)*, pp. 962–965, Hangzhou, Zhejiang, China, October 2014.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, 2005.

[12] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[13] G. Yildiz and B. Dizdaroglu, "Traffic sign detection via color and shape-based approach," in *Proceedings of the 2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–5, Ankara, Turkey, November 2019.

[14] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: optimal speed and accuracy of object detection," 2020.

[15] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," 2018.

[16] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6517–6525, Honolulu, HI, USA, July 2017.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, Las Vegas, NV, USA, June 2016.

[18] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot MultiBox detector," in *Proceedings of the Computer Vision - ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., pp. 21–37pp. 21–, Amsterdam, The Netherlands, October 2016.

[19] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[20] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: a simple and strong anchor-free object detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, p. 1, 2020.

[21] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Santiago, Chile, December 2015.

[22] X. Yang, W. Liu, S. Zhang, W. Liu, and D. Tao, "Targeted attention attack on deep learning models in road sign recognition," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4980–4990, 2021.

[23] S. He, L. Chen, S. Zhang et al., "Automatic recognition of traffic signs based on visual inspection," *IEEE Access*, vol. 9, Article ID 43261, 2021.

[24] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proceedings of the Neural Information Processing Systems, 2017*, pp. 3859–3869, Montreal, Canada, December 2017.

[25] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, "Yolo V4 for advanced traffic sign recognition with synthetic training data generated by various gan," *IEEE Access*, vol. 9, Article ID 97242, 2021.

[26] D. Tabernik and D. Skocaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Transactions on*

Intelligent Transportation Systems, vol. 21, no. 4, pp. 1427–1440, 2020.

[27] K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[28] J. Cao, J. Zhang, and W. Huang, “Traffic sign detection and recognition using multi-scale fusion and prime sample attention,” *IEEE Access*, vol. 9, pp. 3579–3591, 2021.

[29] K. Sun, Y. Zhao, B. Jiang et al., “High-resolution representations for labeling pixels and regions,” 2019, http://arxiv.org/abs/1904.04514.

[30] Y. Cao, K. Chen, C. C. Loy, and D. Lin, “Prime sample attention in object detection,” in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Article ID 11588, Seattle, WA, USA, June. 2020.

[31] K. Xie, S. Ge, Q. Ye, and Z. Luo, “Traffic sign recognition based on attribute-refinement cascaded convolutional neural networks,” in *Lecture Notes in Computer Science*, E. Chen, Y. Gong, and Y. Tie, Eds., vol. 9916, pp. 201–210, Springer International Publishing, Cham, Switzerland, 2016.

[32] Z. Cai and N. Vasconcelos, “Cascade R-CNN: delving into high quality object detection,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, June 2018.

[33] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015, http://arxiv.org/abs/1409.1556.

[34] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June. 2016.

[35] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “Distance-IoU loss: faster and better learning for bounding box regression,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, Article ID 13000, NY, USA, April 2020.

[36] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5987–5995, Honolulu, HI, USA, July 2017.

[37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011–2023, 2020.

[38] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, “BAM: bottleneck attention module,” 2018, http://arxiv.org/abs/1807.06514.

[39] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: convolutional block Attention module,” in *Proceedings of the Computer Vision - ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds., pp. 3–19, Springer International Publishing, Munich, Germany, September 2018.

[40] Q. Hou, D. Zhou, and J. Feng, “Coordinate attention for efficient mobile network design,” in *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Article ID 13717, Nashville, TN, USA, June 2021.

[41] D. Hendrycks and T. G. Dietterich, “Benchmarking neural network robustness to common corruptions and surface variations,” 2019, http://arxiv.org/abs/1807.01697.

[42] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, “Detection of traffic signs in real-world images: the German traffic sign detection benchmark,” in *Proceedings of the 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Dallas, TX, USA, August 2013.

[43] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.

[44] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-sign detection and classification in the wild,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2110–2118, Las Vegas, NV, USA, June 2016.

[45] J. Zhang, M. Huang, X. Jin, and X. Li, “A real-time Chinese traffic sign detection algorithm based on modified YOLOv2,” *Algorithms*, vol. 10, no. 4, p. 127, 2017.

[46] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2019, http://arxiv.org/abs/1711.05101.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[48] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, March 2010.

[49] C. Peng, T. Xiao, Z. Li et al., “MegDet: a large mini-batch object detector,” in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6181–6189, Salt Lake City, UT, USA, June 2018.

[50] A. Nascita, A. Montieri, and G. Aceto, “XAI meets mobile traffic classification: understanding and improving multimodal deep learning architectures,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4225–4246, 2021.