Received 9 August 2022, accepted 2 September 2022, date of publication 6 September 2022, date of current version 1 March 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3204818

APPLIED RESEARCH

Faster Light Detection Algorithm of Traffic Signs Based on YOLOv5s-A2

XU YUAN, ALIFU KUERBAN, YIXIAO CHEN, AND WENLONG LIN
School of Software, Xinjiang University, Ürümqi, Xinjiang 830046, China

Corresponding author: Xu Yuan (814482443@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Contract 61562084, and in part by the Natural Science Foundation of the Autonomous Region of China under Contract 2021D01C082.

ABSTRACT Traffic sign recognition systems have been applied to advanced driving assistance and automatic driving systems to help drivers obtain important road information accurately. The current mainstream detection methods have high accuracy in this task, but the number of model parameters is large, and the detection speed is slow. Based on YOLOv5s as the basic framework, this paper proposes YOLOv5S-A2, which can improve the detection speed and reduce the model size at the cost of reducing the detection accuracy. Firstly, a data augmentation strategy is proposed by combining various operations to alleviate the problem of unbalanced class instances. Secondly, we proposed a path aggregation module for Feature Pyramid Network (FPN) to make new horizontal connections. It can enhance multi-scale feature representation capability and compensate for the loss of feature information. Thirdly, an attention detection head module is proposed to solve the aliasing effect in cross-scale fusion and enhance the representation of predictive features. Experiments on Tsinghua-Tencent 100K dataset (TT100K) show that our method can achieve more remarkable performance improvement and faster inference speed than other advanced technologies. Our method achieves 87.3% mean average precision (mAP), surpassing the original model's 7.9%, and the frames per second (FPS) value is maintained at 87.7. To show generality, we tested it on the German Traffic Sign Detection Benchmark (GTSDB) without tuning and obtained an average precision of 94.1%, and the FPS value is maintained at about 105.3. In addition, the number of YOLOv5s-A2 parameters is about 7.9 M.

INDEX TERMS Traffic sign recognition, data augmentation, path aggregation, attention detection head.

I. INTRODUCTION

As an essential component of advanced driver assistance systems (ADAS) and auto-driving systems (ADS), traffic sign detection is one of the current research hotspots. It requires locating the location of traffic signs and determining their categories in the realistic scene. However, the detection process of traffic signs will be affected by many factors, resulting in the problems of unstable accuracy and long processing time [1].

Traffic sign detection methods can be divided into two categories: the traditional approach based on manual design features and the deep learning method based on a convolutional neural network. Traditional methods can be divided into color-based, shape-based, and multi-feature fusion. The manually designed features extracted from the candidate regions are input into the classifier (e.g.., SVM) [2] to obtain the detection results. However, the feature level of manual design is low and cannot represent diverse objectives well. Therefore, the generalization ability is poor in complex scenarios. In contrast, deep learning methods learn features from many target samples, and a rich convolution hierarchy can represent complex target features. Detection methods based on the convolutional neural network (CNN) are divided into single-stage and two-stage procedures. The two-stage process has high detection accuracy but slow speed [3]. The single-stage process is low in precision but fast [4]. In order to solve problems in image processing, a variety of CNN variants [5] have evolved to solve specific problems. However, computer vision requires a general CNN structure [6] to construct

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

a more lightweight and high-precision architecture. YOLO series algorithms can achieve a real-time detection rate while maintaining factual accuracy [7].

In addition, the commonly used traffic sign detection dataset is the German Traffic Sign Detection Benchmark (GTSDB) [8]. GTSDB simply divides all the classes into three categories due to the small amount of data. Three types of traffic signs do not meet the actual requirements. To address this problem, we choose Tsinghua-Tencent 100K (TT100K) dataset [9] to avoid the above situation. Traffic sign instances of TT100K are mainly small targets. We also test on the GTSDB dataset to validate generality without tuning. Many detection methods [10] cannot solve the problem in real-time without guaranteeing accuracy. We try to find a better balance between detection accuracy and speed with few parameters and computation.

As a part of ADAS, traffic sign detection aims to remind drivers of road conditions and assist drivers in making reasonable decisions quickly in complex road conditions instead of making decisions for drivers. Therefore, we believe the detection speed is more critical to meet the frequent and changeable traffic conditions based on ensuring accuracy. Suppose the model cannot reach the real-time inference speed. In this case, the location of vehicles and traffic signs may have significantly changed when the detection result appears, and the detection result loses its significance in assisting drivers. Taking GTSDB as an example, the detection category is divided into three super-class. When there is a deviation in a specific instance of the detection category, the driver can also make a reasonable decision according to the super-class information of the detection category and road conditions.

In selecting the model, the first is based on the needs of reality. A single-stage model than a two-stage model has fewer parameters and fewer calculations. It still meets the accuracy requirements and is easier to deploy in the vehicle system. Second, the development trend of CNN is no longer to deepen the width and depth of the model to increase the model capacity and improve the model performance. Exploring the nature of the CNN network and eliminating those redundant parts will inevitably make the model more lightweight and efficient. Therefore, this paper takes the single-stage minimum model in the classic YOLOv5 model as an example for improvement.

Our main contributions are as follows:

First, an effective data augmentation strategy is proposed to solve the shortage of category instances and alleviate the imbalance of categories. Based on Mosaic, the target is cut and copied to increase the number of category instances, and at the same time, the marks are cropped, scaled, and arranged randomly.

Second, the path aggregation module is proposed to compensate for the information attenuation caused by the limitations of the Feature Pyramid Network (FPN) and improve multi-scale feature representation capability. The module is done by establishing new propagation paths.

Third, we designed an attention detection head module to replace the original detection head to enhance the representation ability of predictive features. At the same time, the problem of information aliasing in the process of cross-scale fusion is solved.

II. RELATED WORK

Methods based on deep learning have achieved great success in traffic sign detection, and these methods have achieved outstanding results on open datasets. Next, we will discuss the detection methods based on CNN and their application in traffic sign detection.

Zhu *et al.* [9] created the TT100K dataset and proposed a CNN-based model that can detect minor traffic signs effectively. Unlike most previous models, the detection target accounts for a large proportion of the image, which does not work effectively when the detection target accounts for a small proportion of the picture. Lu *et al.* [11] focused on traffic sign detection in street view and proposed a visual attention model network. The network is divided into two parts: one generates candidate areas, and the other conducts positioning and classification in candidate areas. At the same time, the candidate region is generated based on faster R-CNN with low computational cost and high confidence. To deal with the problem of small sign size, Liu *et al.* [12] proposed DR-CNN, which adds a deconvolution layer and a normalized layer before the output convolution layer, and gets the fused feature layer by splicing the features of different layers to provide richer information. A two-stage classification loss function is designed to distinguish positive and negative samples and improve training efficiency. For solving the high-precision real-time detection task of traffic signs in natural scenes, Zhang *et al.* [13] proposed MSA_YOLOv3, which used mixup image augmentation and introduced a multi-scale spatial pyramid pooling module to learn richer features. A bottom-up path is used to enhance FPN by using low-level features for better positioning accuracy. Liang *et al.* [14] proposed a two-stage network. The regional proposal stage adopted the deep feature pyramid structure and strengthened the lateral connection to make the network sensitive to the semantic features of small targets. Feature transfer and multiplexing are enhanced through dense links to obtain a more accurate classification with fewer parameters in the classification stage. Small traffic sign detection is still a challenge. Inspired by YOLOv4 and YOLOv5, Chen *et al.* [15] introduced high-level features to construct detection heads and designed a sensing domain module on the neck of the model to capture the context information of feature maps. Then they adjusted the detection head grid to adapt to small targets and enhanced images by random erasure.

III. METHODS

A. IMPROVED YOLOv5s

Ultralytics Llc proposed YOLOv5s [16] in May 2020. YOLOv5s network can be roughly divided into four parts:

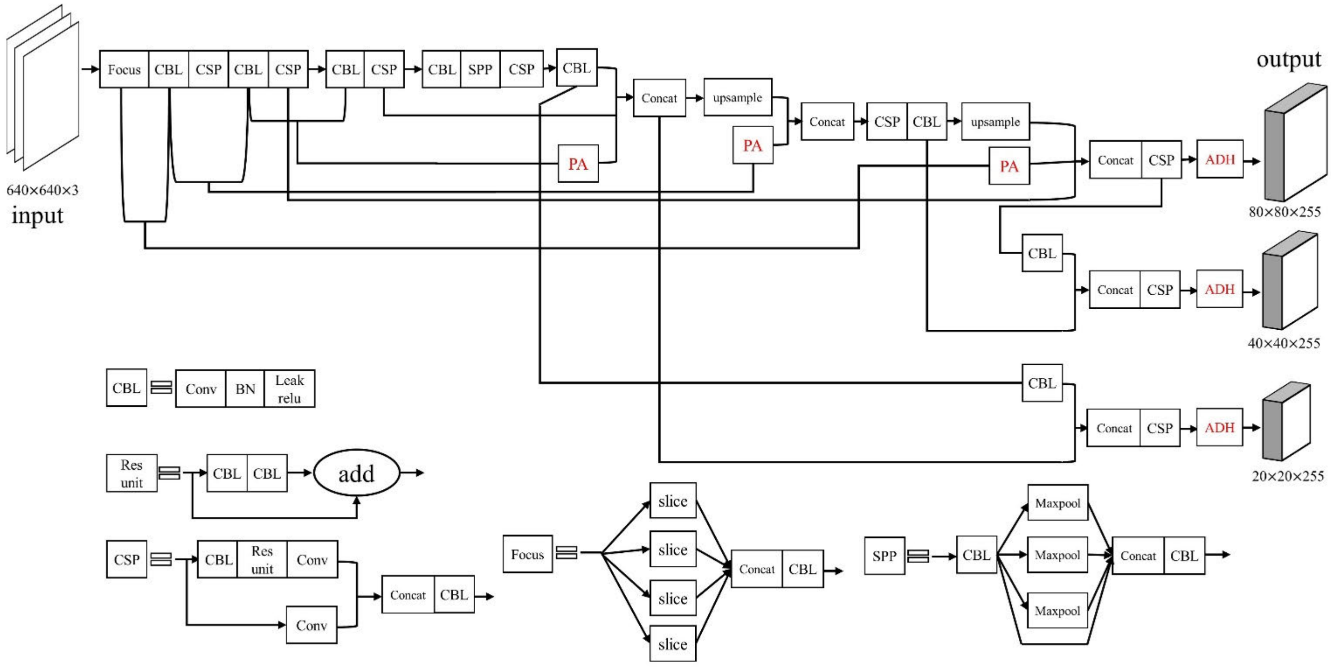


FIGURE 1. The architecture of YOLOv5s-A2. The input is an image scaled to 640 × 640. The output is three feature vectors, as shown in the figure. Conv is a standard convolutional layer, and BN is Batch Normalization. LeakRelu is the activation function. Add denotes the element-wise summation operation. Concat is splicing operations on the channel dimension. PA and ADH are described in detail in the following sections.

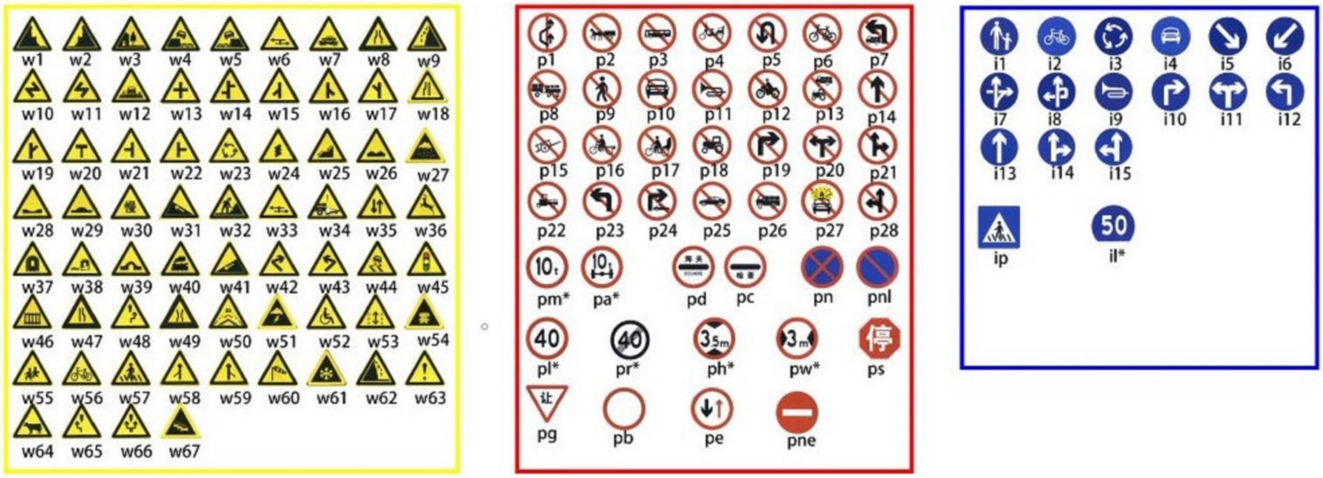

w1 w2 w3 w4 w5 w6 w7 w8 w9
w10 w11 w12 w13 w14 w15 w16 w17 w18
w19 w20 w21 w22 w23 w24 w25 w26 w27
w28 w29 w30 w31 w32 w33 w34 w35 w36
w37 w38 w39 w40 w41 w42 w43 w44 w45
w46 w47 w48 w49 w50 w51 w52 w53 w54
w55 w56 w57 w58 w59 w60 w61 w62 w63
w64 w65 w66 w67
p1 p2 p3 p4 p5 p6 p7
p8 p9 p10 p11 p12 p13 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 2. Traffic signs in TT100K.

input layer, backbone part, neck part, and output layer. Backbone mainly adopts the Focus and Cross Stage Partial Networks (CSP) structures. The Neck part is composed of FPN and Path Aggregation Network (PAN) design to enhance information transmission and maintain the accuracy of spatial information. The output layer adopts the following improvements: Mosaic data enhancement and adaptive Anchor calculation, Generalized Intersection over Union Loss (GIoU_Loss) is used in training, and Distance Intersection over Union (DIoU) is used for prediction box filtering.

The network divides the input images into $S \times S$ grids. If the target center falls into a grid, the grid predicts the target.

This paper improves the model's performance through data augmentation, path Aggregation module of FPN, and attention detection head module. We named the improved model YOLOv5s_A2, as shown in Fig. 1.

B. DATASETS

The experiments on traffic signs mainly used the TT100K [9] dataset, a traffic-sign benchmark from 100K Tencent Street View panoramas.

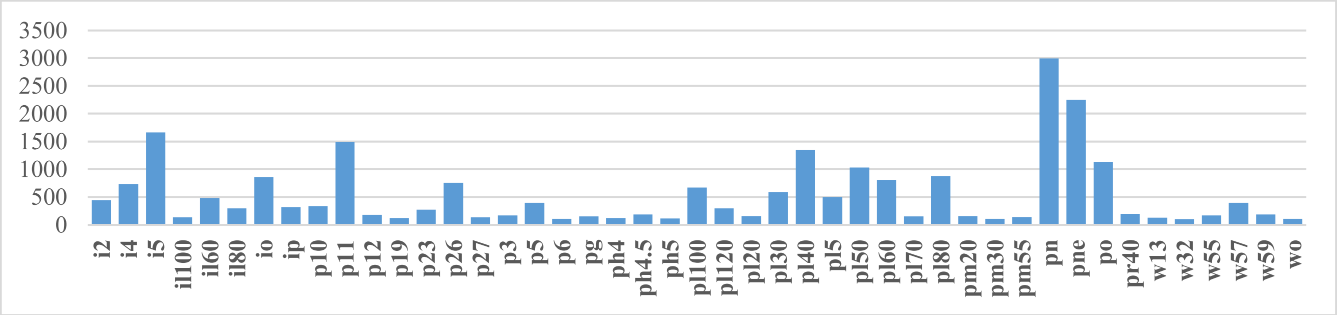
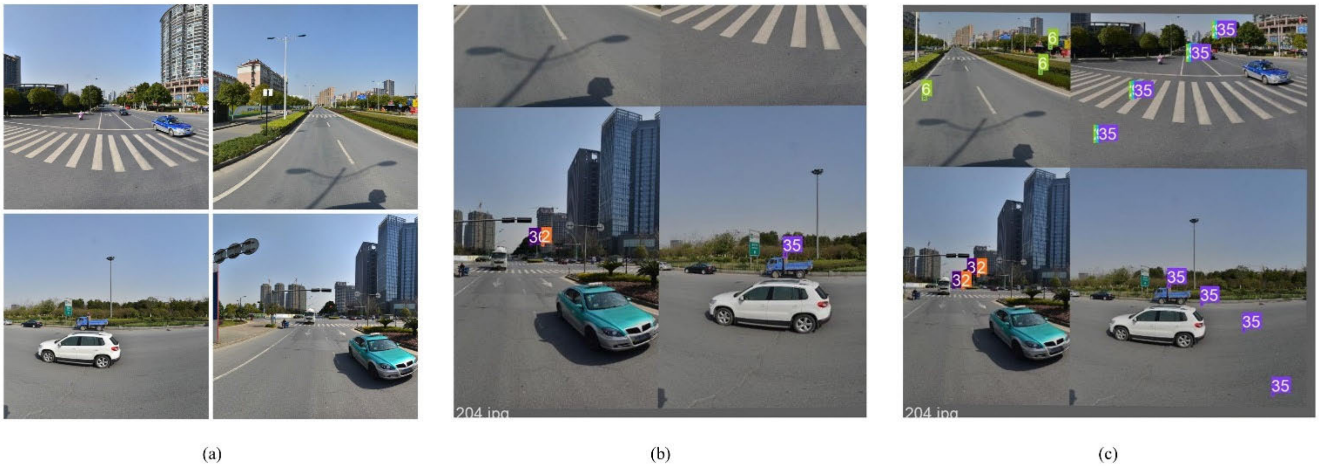


FIGURE 3. The number of class instances in TT100K.



(a) (b) (c)

FIGURE 4. The (a) is an original input image of the model, and (b) is a Mosaic-enhanced image. The (c) is an enhanced image for our method, showing an increase in instances.

TT100K: The dataset contains 9167 images (6105 for training and 3071 for testing). The dataset is very challenging, including many small-sized traffic signs and covering significant changes under different lighting and weather conditions with a size of 2048 × 2048, which can reflect traffic signs in real natural scenes. It can be seen that the areas of the bounding box in the TT100K are mostly smaller than 96 × 96 pixels. The proportion of small, medium and large objects is 40.4%, 52.25%, and 7.35%, respectively. The category instance distribution is shown in Fig. 3.

GTSDB: The GTSDB provides 600 images for training and 300 for testing. The image size in GTSDB is 1360 × 800. GTSDB divides traffic signs into four super-class: prohibitory signs, mandatory signs, danger signs, and others though GTSDB provides 43 classes in total. The proportion of small, medium and large objects is 35.6%, 60.7%, and 3.7%, respectively. The number of training and test set instances is 815 and 353, respectively.

Partition of datasets: Although the TT100k includes 234 categories for traffic signs, we found that some categories appear only in the training dataset or the test dataset in the experiment. Referring to [9], we select 45 types from the TT100K dataset for the traffic sign detection study; the training set to test set is approximately 7: 3, and data augmentation is used in the training set. Other comparison models are tested on the original training set and test set. Referring to [8], we detect three super-class in GTSDB.

C. DATA AUGMENTATION

According to Vicinal Risk Minimization (VRM) [17], the model's generalization ability can be improved by adding data similar to the training samples. This paper proposes an effective data augmentation strategy for insufficient target instances. The data enhancement strategy is a real-time enhancement of the calling picture in the model training process and does not change the size of the dataset. The enhancement result of the same image in different training cycles is not the same. First, there is a significant imbalance in the selected dataset's category instances, as shown in Fig. 3.

We use the idea of the cutmix [18] method as a reference and only apply it to clips to detect the target object rather than the whole image. Unlike using a logo [19] to replace the original detection target composite image enhanced dataset, this method directly takes enhanced images from the original dataset during training and generates multiple target instances. The clipping, scaling, and arranging targets are

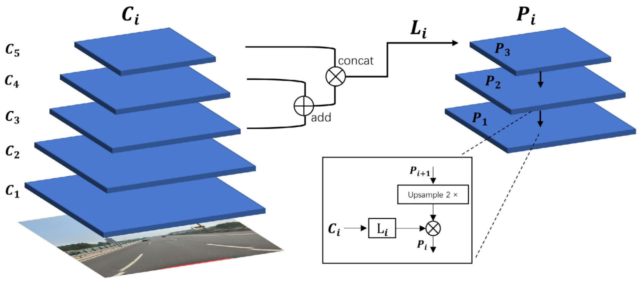
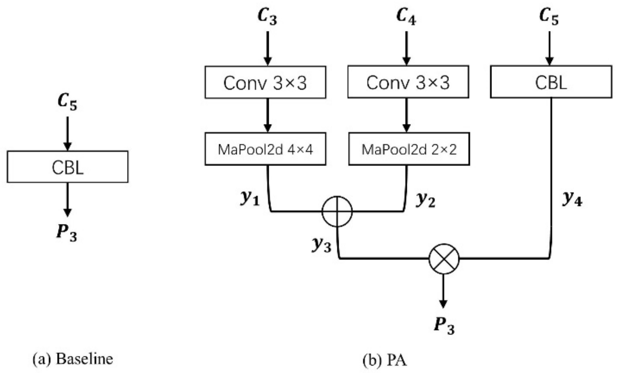


FIGURE 5. The path aggregation module of FPN.



(a) Baseline (b) PA

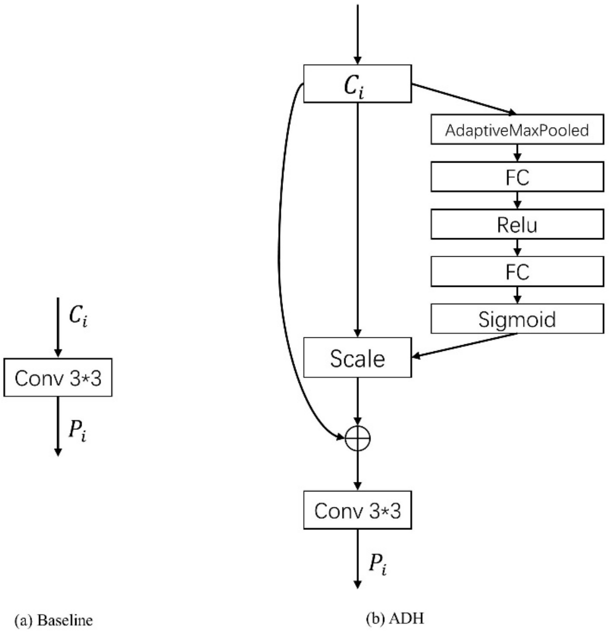
FIGURE 6. The structures of (a) original lateral connection Li and (b) PA module.

random, so our method is more straightforward and effective. According to the situation, we will simulate a target instance in a photo multiple times due to the imbalance of categories. In this paper, we will randomly copy up the target instance in a picture 3 times and place the copied target instance each time.

However, its disadvantage is that the added category instances have the same characteristics as the clipped instances, which cannot increase the diversity of samples. In this regard, we execute random scaling, clipping, arrangement, and small range color transformation for the target instance of copying and pasting. Scaling can generate multi-scale category instance samples, increasing the multi-scale variation of the target sample. Random clipping will cause incomplete target instances, similar to the target being blocked by other objects, making the network focus on the features of the target itself to detect the target object according to the partial elements of the target. The random arrangement will move the target object to any position in the image to enrich the detected object's background information and increase the model's robustness against irrelevant context information. As shown in Fig. 4, the enhanced image has more target instances, and the added target instances are diverse. We combined it with the mosaic [20] strategy operations to achieve better results.

D. PATH AGGREGATION MODULE OF FPN

Recognizing objects of different sizes is a fundamental challenge in computer vision. Although FPN [19] is simple and effective in up-sampling and stacking, the fusion of multiple features with extensive semantic gaps results in a suboptimal feature pyramid. It loses part of feature information in the top-down and bottom-up paths [21] caused by multiple sampling in path propagation, resulting in a decline in multi-scale feature representation.



(a) Baseline (b) ADH

FIGURE 7. Structure (a) is the original detection head, and Structure (b) is the attention detection head.

As shown in Fig. 5, different from the general FPN structure, we aggregate the path of sampling multiple high-resolution feature maps in the bottom-up routes to enhance the spatial information further. The core idea is to increase the number of lateral connections in the encoding and decoding parts of the pyramid structure to improve spatial accuracy further while maintaining high semantic feature representation. We introduce the path aggregation module to enhance FPN and improve feature representation capability.

The baseline lateral connection used in FPN is illustrated in Fig. 6(a), which is a convolution block. Instead of only using one convolution block in FPN, we enhance the positioning and feature abstraction capabilities of FPN by aggregating multi-scale features from different layers paths. As shown in Fig. 6(b), we added high-level connections C_3 and C_4 from the encoder to the decoder.

However, adding lateral connections raises new issues. High-resolution feature maps need to be adjusted to meet the spatial dimensions of the connected layers. Firstly, we keep the feature map's size unchanged by using the padding operation in the 3 × 3 convolutional layer. Then, we adjust the feature map to a uniform size through a 2D pooling operation of specific convolution kernel size to reach the end of the y_1, y_2 path. C_5 goes through the original convolution block to reach the end of y_4. And then, we merge y_1 and y_2 to

TABLE 1. Ablation studies of the improvement strategy.

Improvement		Experiment1	Experiment2	Experiment3	Experiment4	Experiment5	Experiment6
DATA_AUG			√			√	√
PA				√		√	√
ADH					√		√
Dataset		TT100K/GTSDB	TT100K/GTSDB	TT100K/GTSDB	TT100K/GTSDB	TT100K/GTSDB	TT100K/GTSDB
Evaluation indicator	Map/%	79.4/90.6	84.2/91.7	81.8/91.5	80.2/92.6	87/92	87.3/94.1
	Recall/%	75.2/86.5	79.7/86.7	76.4/84.4	76.5/86.1	80.1/86.5	81.9/90.5
	FPS/(f/s)	117.6/126.6	117.6/126.6	100.0/107.5	111.1/113.6	92.3/112.4	87.7/105.3

TABLE 2. Comparison of each method on TT100K.

Method	Map/%	Resolution	FPS/(f/s)	Parameters
Faster R-CNN	52.9		13.1	
SSD	44.6	512 × 512	28.9	
MAS_YOLOv3	86.3	544 × 544	23.8	62 M+
CAB	78	512 × 512	27.9	
YOLOv4	88.4	608 × 608	65.8	30 M
YOLO v5-l	86.8	608 × 608	71.9	44 M
YOLOv5-x	88.1	608 × 608	58.4	83 M
TSR-SA	90.2	608 × 608	48.8	30 M+
YOLOv5s	79.4	640 × 640	117.6	6.7 M
YOLOv5s-A2	87.3	640 × 640	87.7	7.9 M

get y_3 with the element-wise summation operation. Finally, y_3 and y_4 are fused to get P_3 by splicing operations on the channel dimension. We introduce low-level features in the P_3 layer to enhance the information of features. The P_3 can be represented as follows,

$$P_3 = [y_3, y_4] \tag{1}$$

$$y_1 = f_1(C_3) \tag{2}$$

$$y_2 = f_2(C_4) \tag{3}$$

$$y_3 = y_1 + y_2 \tag{4}$$

$$y_4 = f_3(C_5) \tag{5}$$

where f_1 denotes the 3 × 3 convolutional layer and 4 × 4 maxPooled2d, f_2 denotes the 3 × 3 convolutional layer and 2 × 2 maxPooled2d, f_3 denotes the 3 × 3 convolutional layer, batch normalization and LeakRelu activation function. The P_3 can be represented as follows,

$$P_3 = [f_1 (C_3) + f_2(C_4), f_3(C_5)] \tag{6}$$

We did the same thing with P_1 and P_2, but with some differences from P_1. P_2, P_1 can be represented as follows,

$$P_2 = [f_1 (C_2) + f_2 (C_3), f_3 (C_4), f_4 (P_3)] \tag{7}$$

$$P_1 = [f_1 (C_1) + f_2 (C_2), f_3 (C_3), f_4 (P_2)] \tag{8}$$

where f_4 denotes the 2 × upsampling operations. P_1, P_2 and P_3 of the original model are expressed as $P_{1'}$, $P_{2'}$ and $P_{3'}$; those can be represented as follows,

$$P_{3'} = f_3(C_5) \tag{9}$$

$$P_{2'} = [f_3 (C_4), f_4 (P_3)] \tag{10}$$

$$P_{1'} = [f_3 (C_3), f_4 (P_2)] \tag{11}$$

We name the path aggregation module of FPN PA. Compare Eq.(6), Eq.(7), Eq.(8) with Eq.(9), Eq.(10), Eq.(11), P_1, P_2, P_3 is improved in multi-scale feature representation ability, which only needs a small amount of parameters cost.

E. ATTENTION DETECTION HEAD MODULE

FPN structure [22] has an aliasing effect in cross-scale fusion. Multi-scale fusion and cross-layer connection are widely used to improve model performance. However, the direct fusion confuses the localization and recognition information in the output feature tensor due to the semantic difference in the multi-scale feature maps.

The model divides the input image into HW grids, and each grid predicts 3 anchor boxes. H, W represent the height and width of the feature map. Each anchor box requires (x, y, w, h, confidence) 5 basic parameters and C category probabilities. The model output shape is $HW \times (C + 1 + 4) \times 3$. As shown in Fig. 7(a), the original model uses a 3 × 3 convolution as the detection head. Its incomplete feature representation will lead to cross confusion of prediction data of different anchor frames. Therefore, we designed an attention detection head

TABLE 3. Comparison of map of each class on TT100K.

Method	i2	i4	i5	il100	il60	il80	io	ip	p10
MSA_YOLOv3	85.0	84.0	92.0	85.0	95.0	89.0	85.0	90.0	74.0
YOLOv4	82.4	93.7	96.5	89.1	99.8	98.6	84.5	82.5	85.3
YOLOv5-x	88.6	94.7	97.1	94.3	98.6	94.9	90.6	95.8	81.9
TSR-SA	88.9	93.1	96.6	99.9	99.9	97.0	85.6	87.7	84.6
ours	87.3	94.5	**97.4**	93.7	98.2	92.6	85.3	89.5	80.8

Method	p11	p12	p19	p23	p26	p27	p3	p5	p6
MSA_YOLOv3	72.0	78.0	73.0	82.0	81.0	83.0	81.0	77.0	72.0
YOLOv4	79.6	94.5	89.3	85.6	90.1	95.8	89.3	96.7	86.0
YOLOv5-x	87.9	83.9	82.7	92.7	92.1	94.3	93.2	91.3	80.8
TSR-SA	90.5	90.1	81.1	92.6	93.5	99.1	95.9	97.4	68.5
ours	83.3	77.1	84.8	90.5	91.8	90.9	85.1	**97.6**	77.6

Method	pg	ph4	ph4.5	ph5	pl100	pl120	pl20	pl30	pl40
MSA_YOLOv3	92.0	82.0	83.0	63.0	88.0	80.0	75.0	74.0	74.0
YOLOv4	86.5	85.3	86.3	86.0	94.6	91.5	86.3	86.0	88.2
YOLOv5-x	96.1	80.6	92.7	77.6	95.6	93.2	80.2	87.7	88.6
TSR-SA	83.7	84.2	88.5	91.8	93.0	96.1	81.9	88.1	93.3
ours	93.7	77.8	82.4	74.7	95.2	93.9	81.0	**89.1**	92.8

Method	pl5	pl50	pl60	pl70	pl80	pm20	pm30	pm55	pn
MSA_YOLOv3	78.0	72.0	76.0	79.0	72.0	76.0	67.0	71.0	83.0
YOLOv4	92.3	83.8	89.2	87.2	90.6	93.2	78.9	94.7	85.9
YOLOv5-x	89.0	86.7	88.2	80.2	91.5	89.5	82.8	94.7	91.6
TSR-SA	94.0	87.0	89.9	90.9	92.9	89.8	80.4	94.5	94.3
ours	91.2	**90.4**	**92.3**	78.9	90.1	**95.6**	**89.3**	89.4	92.4

Method	pne	po	pr40	w13	w32	w55	w57	w59	wo
MSA_YOLOv3	96.0	76.0	85.0	70.0	91.0	79.0	85.0	73.0	53.0
YOLOv4	95.6	78.4	99.1	83.9	59.5	90.4	96.4	89.7	69.1
YOLOv5-x	98.1	79.6	97.7	76.6	85.6	86.0	89.0	71.9	55.7
TSR-SA	94.5	81.9	97.1	79.5	71.2	92.7	93.7	91.0	68.6
ours	97.4	80.1	95.6	72.5	81.7	80.6	86.1	81.4	62.7

to solve this problem named ADH. From the output shape of the model, we know that the prediction information of the anchor box exists in the channel dimension, so we use channel attention to limit the prediction information.

We intend to use Hou *et al.* proposed coordinate factorizes channel attention as CA [23], an improved version of Squeeze-and-Excitation (SE) published in 2021. It then needs to be used in conjunction with a specific residual structure. Li *et al.* [24], inspired by SE's global average pooling operation, designs different attention units and weights different attention vectors. These advanced modules require more conditions to function. Because the model used in this paper does not meet these conditions, the cost of meeting these conditions is outweighed by the benefits of using higher levels of attention. We have introduced the more commonly used SE [25], which shows a better effect in the model used in this paper.

As shown in Fig. 7(b), the ADH first calculates channel weights according to input features and then performs weighted processing on the input channels. We then avoid possible information loss by introducing a residual link. Finally, it is sent into a 3 × 3 convolutional layer to obtain the final prediction P_i. P_i can be represented as follows,

$$P_i = f_6(f_5\left(C_i\right) + C_i) \tag{12}$$

where f_5 denotes the sequential operations of the Adaptive-MaxPooled layer, full connection layer, Relu activation function, full connection layer, and Sigmoid activation function.

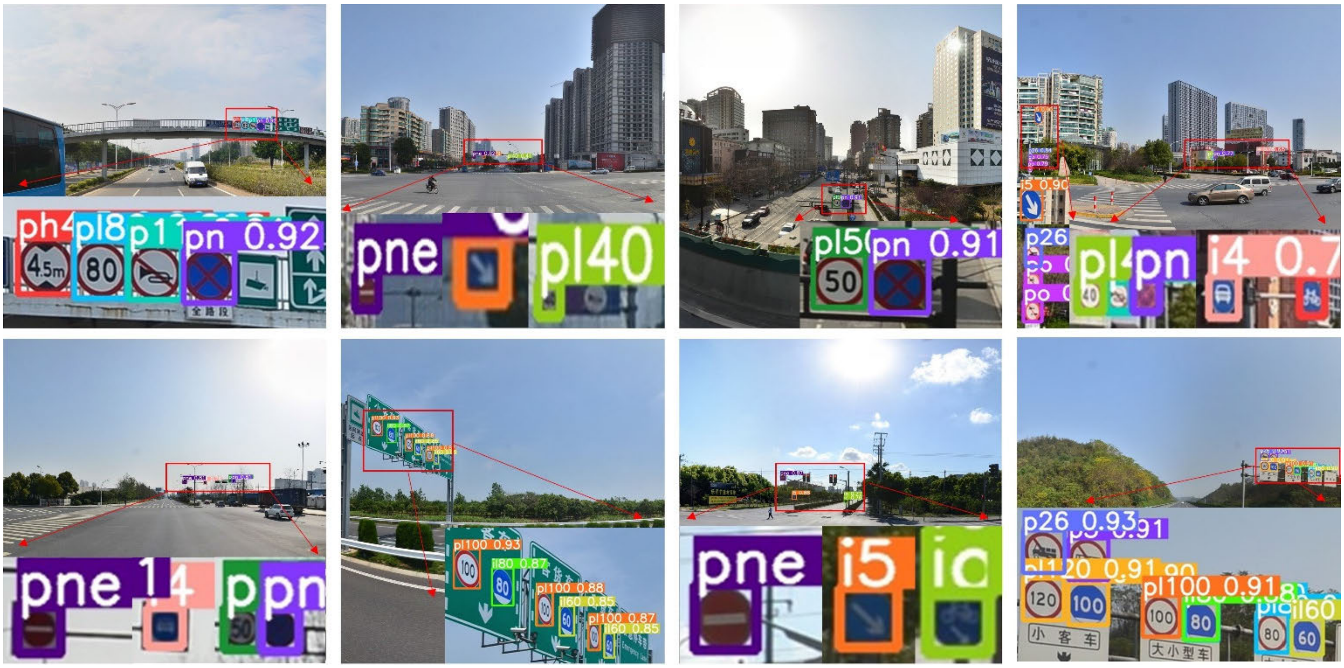


FIGURE 8. Detection renderings of the improved algorithm model in different scenes. We zoomed in on the detection target for better display.

f_6 denotes the 3 × 3 convolutional layer. C_i denotes input feature maps.

The f_5 represents the compression and activation of feature information. The original module used average pooling to generate channel information during the compression phase, while we used maximum pooling to obtain texture information at the expense of background information. Then we introduce a residual connection to combine C_i and f_5 output, and the resulting feature map contains texture and background information. After f_6, the enhanced prediction features are obtained. As shown in Fig. 7(a), the P_i of the original model is expressed as $P_{i'}$. It can be represented as follows,

$$P_{i'} = f_6(C_i) \tag{13}$$

From Eq.(12) and Eq.(13), it is clear that P_i has a more robust feature representation than $P_{i'}$. We replaced all the detection heads, as shown in Fig. 1.

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

The experimental environment is CUDA11.2 + Pytorch1.8.1 + Python3.8, and the practical device is NVIDIA GeForce RTX 3060 Laptop GPU with 6 GB memory. For the hyperparameter setting, the initial learning rate is 0.01, the learning rate strategy is cosine, the training period is 300, the initial input size of the model is 640 × 640, and the Batch size is 16. Pre-training weights are not used during model training. The mean average precision (mAP) is used as the evaluation measurement in this paper. This paper uses a fixed Intersection-over-Union (IoU) value of 0.5 for computing mAP.

B. ABLATION STUDIES

We study the effect of different parts of the modification. All experiments were carried out on the TT100K dataset and GTSDB dataset. As shown in Table 1, DATA_AUG represents the data augmentation strategy, PA represents a path aggregation of the FPN module, and ADH represents an attention detection head module.

Table 1 shows that the mAP improves by 7.9% when DATA_AUG, PA, and ADH are used in TT100K. DATA_AUG belongs to data preprocessing and does not increase the cost of reasoning. PA and ADH can reduce the inference speed but improve the model's performance. We found that the model's performance was enhanced when all changes were used in sequence, indicating that the module changes did not conflict. It achieves 94.1% mAP in GTSDB, which surpasses the original model by 3.5%. It proves that the model's improvement is effective on other datasets, not just tuning for a single dataset, and demonstrates the generalization ability of the improved model. The parameter size of the original model is 6.7 M, and the parameter size of the enhanced model is 7.9 M. About 1.2 M increases the parameter number, but the performance is greatly improved with real-time reasoning speed. The improved model can reach 87.7 f/s and 105.3 f/s on TT100K and GTSDB, respectively, as shown in Table 1. The improved model is still a lightweight, fast model.

C. DETECTION PERFORMANCE AND EFFICIENCY

As shown in Table 2, we compare Faster R-CNN, SSD, MSA_YOLOv3 [13], YOLOv4 [20], YOLOv5x, YOLOv5-L [16], CAB [26], TSR-SA [15] and our model on TT100K. The number of parameters of some models is not provided in the original paper. TSR-SA method adds some modules based on YOLOv4, so the number of parameters is greater than 30 M. It can also be inferred that the number of MSA_YOLOv3 model parameters exceeds 62 M. Compared with the TSR-SA model (published in 2022), the map of YOLOv5s-A2 is 2.9% lower than the highest value. However, the number of parameters is only 26.3%, and the inference speed is 1.8 times that. Compared with other models cited in the article, the superior performance of YOLOv5s-A2 is highlighted. The amount of calculation is positively correlated with the number of parameters, so our model has the least amount of calculation. The reasoning speed cannot be based on the amount of theoretical calculation because of the degree of parallelism, which explains the relationship between the number of model parameters and the reasoning speed in Table 2. It shows that we have achieved high-precision and rapid detection of traffic signs with the minimum number of parameters and computation. In addition, we compare each mAP of the selected category in the TT100K dataset.

As shown in Table 3, we found that the mAP of some categories was low, and a similar situation also occurred in other models. For example, the mAP value of W13 and W55 is 72.5% and 80.6%, which was much lower than the accuracy of different categories. We compared the number of instances and marks differences in Fig. 2 and Fig. 3; the number of instances of W13 and W55 categories is 129, 169. We believe this may be due to the insufficient number of category instances; models tend to learn classes with more samples. Secondly, network features have the insufficient capability to represent complex flags. As shown in Fig. 8, that shows the detection effect of the YOLO5s-A2 for multi-scale targets in the image.

V. CONCLUSION

First, the proposed data augmentation strategy increases the number of category instances and can effectively improve the model performance without sacrificing the inference speed. Second, the PA module builds new propagation paths at the cost of a small number of parameters and reasoning speed and improves the model's performance by improving multi-scale feature representation and reducing feature information loss. Third, ADH can eliminate the aliasing effect of cross-scale fusion and improve the representation ability of prediction features, thus improving performance. Experimental results on TT100K demonstrate that YOLOv5s-A2 is a fast, lightweight and high-precision model.

Mainstream detection methods and open datasets focus on daytime and normal weather conditions. Only a few ways focus on traffic sign detection at night and under bad weather conditions. In the future, we will focus on the performance of traffic sign detection at night and in bad weather conditions and strive to reduce storage and computing requirements.

REFERENCES

[1] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, "F-Cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds," in *Proc. 4th ACM/IEEE Symp. Edge Comput.*, Nov. 2019, pp. 88–100.

[2] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, Jul. 2016, doi: 10.1109/TITS.2015.2482461.

[3] Y. Zhu, M. Liao, W. Liu, and M. Yang, "Cascaded segmentation-detection networks for text-based traffic sign detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 1, pp. 209–219, Jan. 2018, doi: 10.1109/TITS.2017.2768827.

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

[5] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, pp. 228–249, 2021.

[6] C. Patel, D. Bhatt, U. Sharma, R. Patel, S. Pandya, K. Modi, N. Cholli, A. Patel, U. Bhatt, M. A. Khan, S. Majumdar, M. Zuhair, K. Patel, S. A. Shah, and H. Ghayvat, "DBGC: Dimension-based generic convolution block for object recognition," *Sensors*, vol. 22, no. 5, p. 1780, Feb. 2022.

[7] L. Wang, K. Zhou, A. Chu, G. Wang, and L. Wang, "An improved light-weight traffic sign recognition algorithm based on YOLOv4-tiny," *IEEE Access*, vol. 9, pp. 124963–124971, 2021, doi: 10.1109/ACCESS.2021.3109798.

[8] 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 *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Dallas, TX, USA, Aug. 2013, pp. 1–8.

[9] 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)*, Las Vegas, NV, USA, Jun. 2016, pp. 2110–2118.

[10] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, "Deep detection network for real-life traffic sign in vehicular networks," *Comput. Netw.*, vol. 136, no. 8, pp. 95–104, May 2018, doi: 10.1016/j.comnet.2018.02.026.

[11] Y. Lu, J. Lu, S. Zhang, and P. Hall, "Traffic signal detection and classification in street views using an attention model," *Comput. Vis. Media*, vol. 4, no. 3, pp. 253–266, Sep. 2018.

[12] Z. Liu, D. Li, S. S. Ge, and F. Tian, "Small traffic sign detection from large image," *Int. J. Speech Technol.*, vol. 50, no. 1, pp. 1–13, Jan. 2020, doi: 10.1007/s10489-019-01511-7.

[13] H. Zhang, L. Qin, J. Li, Y. Guo, Y. Zhou, J. Zhang, and Z. Xu, "Real-time detection method for small traffic signs based on Yolov3," *IEEE Access*, vol. 8, pp. 64145–64156, 2020, doi: 10.1109/ACCESS.2020.2984554.

[14] Z. Liang, J. Shao, D. Zhang, and L. Gao, "Traffic sign detection and recognition based on pyramidal convolutional networks," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 6533–6543, Jun. 2020, doi: 10.1007/s00521-019-04086-z.

[15] J. Chen, K. Jia, W. Chen, Z. Lv, and R. Zhang, "A real-time and high-precision method for small traffic-signs recognition," *Neural Comput. Appl.*, vol. 34, no. 3, pp. 2233–2245, Feb. 2022, doi: 10.1007/s00521-021-06526-1.

[16] J. Fang, Q. Liu, and J. Li, "A deployment scheme of YOLOv5 with inference optimizations based on the triton inference server," in *Proc. IEEE 6th Int. Conf. Cloud Comput. Big Data Anal. (ICCCBDA)*, Chengdu, China, Apr. 2021, pp. 441–445.

[17] V. N. Vapnik, "The vicinal risk minimization principle and the SVMs," in *The Nature of Statistical Learning Theory*, New York, NY USA: Springer, 2000, pp. 267–290.

[18] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, (Seoul) Korea, Oct. 2019, pp. 6023–6032.

[19] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools Appl.*, vol. 79, pp. 18201–18219, Mar. 2020.

[20] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13029–13038.

[21] C. I. Patel, S. Garg, T. Zaveri, and A. Banerjee, "Top-down and bottom-up cues based moving object detection for varied background video sequences," *Adv. Multimedia*, vol. 2014, pp. 13–33, Jan. 2014.

[22] Y. Luo et al., "CE-FPN: Enhancing channel information for object detection," *Multimedia Tools Appl.*, vol. 81, no. 21, pp. 30685–30704, 2022.

[23] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, Tennessee, Jun. 2021, pp. 13713–13722.

[24] W. Li, K. Liu, L. Zhang, and F. Cheng, "Object detection based on an adaptive attention mechanism," *Sci. Rep.*, vol. 10, no. 1, pp. 1–13, Dec. 2020.

[25] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7132–7141.

[26] L. Cui, P. Lv, X. Jiang, Z. Gao, B. Zhou, L. Zhang, L. Shao, and M. Xu, "Context-aware block net for small object detection," *IEEE Trans. Cybern.*, vol. 52, no. 4, pp. 2300–2313, Apr. 2022, doi: 10.1109/TCYB.2020.3004636.

XU YUAN was born in Shanxi, China, in 1998. He received the B.S. degree in material forming and control engineering from the Guangdong University of Technology, in 2020. He is currently pursuing the M.S. degree with the School of Software Engineering, Xinjiang University. His current research interests include applying neural networks in deep learning and target detection.

ALIFU KUERBAN received the bachelor's degree in computer science from The Central University for Nationalities, in 1993, and the master's degree in software engineering from the University of Electronic Science and Technology of China, in 2010. He is currently a Professor with the School of Software, Xinjiang University. His research interests include natural language processing, image recognition and processing, virtual technology, and computer application technology.

YIXIAO CHEN was born in Guangdong, China, in 1997. He received the B.S. degree in computer science and technology from Guangdong Ocean University, in 2020. He is currently pursuing the M.S. degree in software engineering with Xinjiang University. His current research interests include applying neural networks in deep learning and target detection.

WENLONG LIN was born in Fujian, China, in 1996. He received the B.S. degree in machine design from Shihezi University, in 2019. He is currently pursuing the M.S. degree in software engineering with Xinjiang University. His current research interests include applying neural networks in deep learning and image processing.