IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 19 October 2023, accepted 6 November 2023, date of publication 8 November 2023,
date of current version 15 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331426

RESEARCH ARTICLE

Improved Traffic Sign Detection Model Based on YOLOv7-Tiny

FEIFAN SHE, ZHIYONG HONG, ZHIQIANG ZENG, AND WENHUA YU
Facility of Intelligence Manufacture, Wuyi University, Jiangmen, Guangdong 529000, China

Corresponding author: Zhiyong Hong (hongmr@163.com)

This work was supported in part by the Joint Research and Development Fund of Wuyi University, Hong Kong, and Macau, under Grant 2019WGALH21 and Grant 2022WGALH18; and in part by the National Natural Science Foundation of China under Grant 52305550.

ABSTRACT Traffic sign detection is a critical task in the autonomous driving. Ordinary networks cannot obtain satisfactory results in traffic sign detection because the size distribution of traffic signs are extremely unbalanced. To overcome this challenge, this paper proposed an improved YOLOv7-Tiny object detection model. Firstly, a path connection strategy was proposed to enhance small-scale feature representation. Compared to the original FPN connection strategy, it adds a path that leads out of the backbone and connects into the Feature Pyramid Network(FPN). Secondly, we proposed a new down-sampling module—–Slice-Sample. By slicing, the size of the feature map is reduced and subsequently, the weights of the sliced feature map channels are assigned using the channel attention mechanism. It can reduce the loss of feature information. Additionally, a module for detecting attention was proposed to address the aliasing effect found in the fusion of different scales. This channel attention mechanism not only focuses on the correlation of neighboring channels, but also employs two branches to increase the model's ability to extract information from the feature map. Experiments on the German Traffic Sign Detection Benchmark (GTSDB) showed that the improved model can achieve more remarkable performance than yolov7-tiny. Our method achieved 93.47% mean average precision (mAP) surpassing the yolov7-tiny's 7.48%, and the frames per second (FPS) value is maintained at 67.5. Besides, our method is superior to other lightweight models on the GTSDB. To demonstrate the generalizability of our approach, we tested it on the Tsinghua-Tencent 100K dataset (TT100K) without tuning and obtained 66.29% mAp surpassing the yolov7-tiny's 7.59%. In addition, the number of parameters of improved YOLOv7-Tiny is about 23.29 M.

INDEX TERMS Traffic sign detection, feature pyramid network, down-sampling, attention mechanism.

I. INTRODUCTION

Traffic sign detection is a widely studied area of research, focusing on identifying and classifying traffic signs in real-world scenarios. However, the detection process is susceptible to various factors that can affect the accuracy and processing time, resulting in unstable performance [1].

In general, the detection methods for traffic signs can be divided into two categories: the conventional method, which relies on manual design features, and the deep learning method, which is based on Convolutional Neural Network (CNN). The manual design features have limitations in effectively representing diverse objectives, which leads to poor generalization ability in complex scenarios. In contrast, CNN models have the ability to learn features from a large number of samples and can represent complex object features through a rich convolution hierarchy. Consequently, many scholars currently employ CNN models for object detection.

Detection methods, which rely on the CNN, can be categorized into two types: single-stage and two-stage algorithms. The single-stage algorithms, unlike the two-stage algorithms, don't produce candidate regions. Some well-known examples of the single-stage algorithms include the YOLO [2], [3], [4], [5], [6], [7] and the SSD [8], [9], [10] series. Conversely, the two-stage algorithms necessitate the generation of candidate regions. An exemplary representative of the two-stage algorithms is the R-CNN [11], [12], [13] series.

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

The above generalized methods are not very effective in traffic sign detection. Therefore, some scholars have designed models for traffic sign detection. Tabernik and Skočaj [14] used Masked R-CNN to handle the entire process of detection and recognition. They solved the problem of detecting and recognizing a large number of traffic sign categories by performing end-to-end automatic learning. Furthermore, they expanded their traffic sign detector by incorporating data augmentation and Online Hard-Example Mining (OHEM) [15] to further enhance its performance. Ruta et al. [16] opted for a simple yet resilient image representation constructed atop the Colour Distance Transform(CDT). Building on this representation, they introduced an algorithm for selecting features that capture a varying-size collection of local image regions, ensuring maximum dissimilarity between each specific sign and all other signs. In contrast, Ren et al. [17] opted to replace a selective search algorithm with a Region Proposal Networks (RPN) technique, allowing for the extraction of region suggestions and facilitating end-to-end computation for object detection. This modification significantly boosts the overall efficiency of the detection process by leveraging shared convolutional layers. Together these studies provide important insights into the collection of image regions of different sizes. However, they all choose two-stage algorithms for improvement, so the accuracy of detection is high but the real-time performance of the network is poor.

Some other scholars have improved the detection accuracy by increasing the complexity of the network. Li et al. [18] introduced the Perceptual Generative Adversarial Network (Perceptual GAN) model, which aims to minimize the dissimilarity in representation between small and large objects. Their model exhibited excellent performance when evaluated using the TT100K dataset. Rehman et al. [19] devised a novel and effective approach utilizing discriminating patches (d-patches). They put forth a method that enhances the d-patches through the integration of vocabulary learning characteristics, enhancing their ability to handle robust occlusions. Zhu et al. [20] utilized a fully convolutional network (FCN) [21] to identify potential regions of traffic signs. Subsequently, a CNN was employed to classify the detected regions. While this approach yielded satisfactory results, the computational cost was high due to the utilization of FCN. These studies have made important contributions to improving the accuracy of traffic sign detection, but their methods lead to overly complex models with a large number of parameters.

In order for the model to have excellent real-time performance, some scholars have streamlined the network model. Zhang et al. [22] introduced two innovative, lightweight networks that can achieve higher precision in detection while maintaining fewer trainable parameters in the model. Another approach proposed by Lu et al. [23] involved the use of two sub-networks for traffic sign detection. The Attention Proposal Modeler (APM) was used to identify regions likely to contain traffic signs, followed by the Accurate Locator and Recognizer (ALR) to localize and classify the signs in these regions. Their approach through knowledge distillation reduces the number of parameters of the model, but the accuracy of the model for traffic sign detection is reduced.

Some scholars have focused on the single-stage algorithm while keeping the number of parameters of the model low as well as the detection speed high. Cai et al. [24] proposesd a one-stage object detection framework for improving the detection accuracy based on the YOLOv4. They adopted the CSPDarknet53_dcn(P) as the backbone network. Chen et al. [25] devised a module placed on the model's neck that captures contextual information from the feature maps, known as a sensing domain module. Tian et al. [26] developed a multi-scale recurrent attention network, encompassing both a multi-scale attention module and a recurrent attention module. Yuan and his team [27], designed the implementation of a network merging convolution and de-convolution layers, aiming to enhance the feature maps and simultaneously extract higher-level semantic features. Liu et al. [28] used LPFA-Conv module to enhance the sensory field of the detection head based on Yolov5. In addition, they replaced the original loss function by utilizing the Wasserstein Distance(NWD) to enhance the detection ability of the model for small objects. These studies have inspired us by improving the single-stage network's ability to detect traffic signs by replacing the lightweight backbone, adding the attention mechanism, replacing the downsampling module, and improving the loss function. However, their methods do not have enough feature extraction for small objects.

The detection of small traffic signs in real scenes is a challenging task, as ordinary networks struggle to achieve satisfactory results due to the unbalanced size distribution of traffic signs. To address this issue, we propose a new detection model based on YOLOv7-Tiny, which aims to improve the efficiency of detecting small signs. Our main contributions can be summarized as follows:

(1)In order to effectively utilize the precise location information at the bottom of the backbone network, we design an enhancement path that enhances the feature pyramid structure from the bottom up. Compared with the original FPN connection strategy, it adds a path leading from the backbone bottom layer and connecting to the FPN, which allows the small object information in the bottom layer of the backbone to be fused into the FPN. This approach aims to improve the accuracy of locating small objects in high-resolution images.

(2)Inspired by the residual structure and the feature slicing, we proposed a down-sampling module(named Slice-Sample)to reduce information loss during downsampling. It reduces the size of the feature map by slicing, and then assigns the channels of the sliced feature map through the channel attention mechanism. This approach does not need to increase the number of neurons in the neural network compared to convolution and avoids the loss of information compared to pooling.

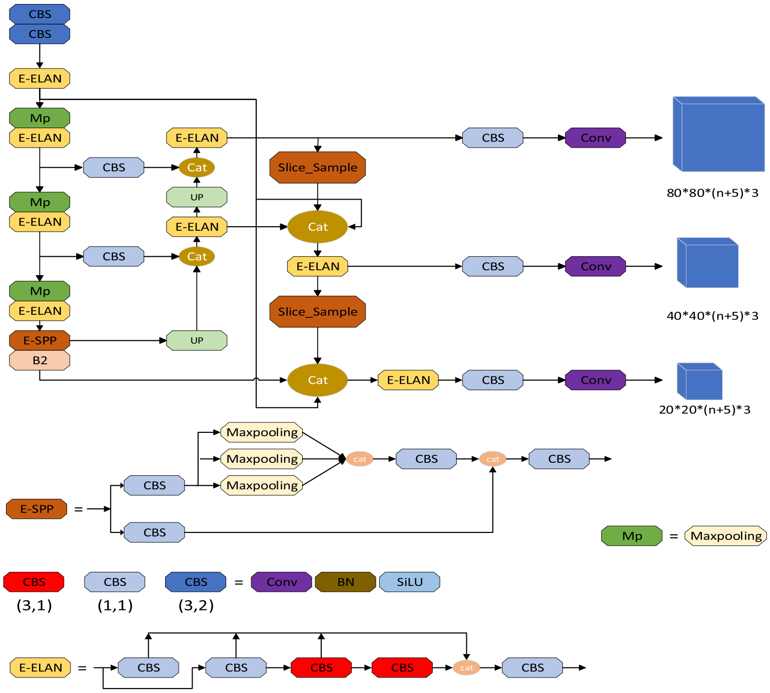


FIGURE 1. The structure of improved YOLOv7-tiny.

(3)The B2 (two-branch) channel attention mechanism was proposed for better integration of information in downsampling module. This channel attention mechanism not only focuses on the correlation of neighboring channels, but also employs two branches to increase the model's ability to extract information from the feature map. Therefore, it is not only used as a part of Slice-sample, but also has a performance improvement over the original backbone when combined with the backbone in the network.

(4)To improve the real-time performance of the network, the EIOU loss function was introduced to reduce the optimization error of the network.

Experiments showed that our methods can effectively improve the detection speed and accuracy on GTSDB and TT100K.

The remainder of this paper is organized as follows: We discuss the detection framework in SectionII. SectionIII presents the experimental results, and SectionIV concludes the paper.

II. THE IMPROVEMENT OF YOLOv7-TINY MODEL

The YOLOv7-tiny is a small model of the yolov7 [29] family, with a parameter count of only 6.2 M. The standard yolov7-tiny model's structure consists of four main parts:

(1)Input module: the primary function of the input module is to resize the input image to a predetermined size, fulfilling the size criteria of the backbone. The images are pre-processed via several operations such as data augmentation within this module.

(2)Backbone: the backbone of yolov7-tiny consists of CBS(Conv+Bn+Silu) layer, MP(Max Pooling) layer.

(3)Feature Pyramid Network: YOLOv7-tiny is the same as YOLOv5 network and also adopts the traditional Path Aggregation Feature Pyramid Network(PAFPN).

(4)Detection head: in the detection head part, YOLOv7-tiny chooses the IDetect detection head that indicates 3 object sizes: large, medium and small.

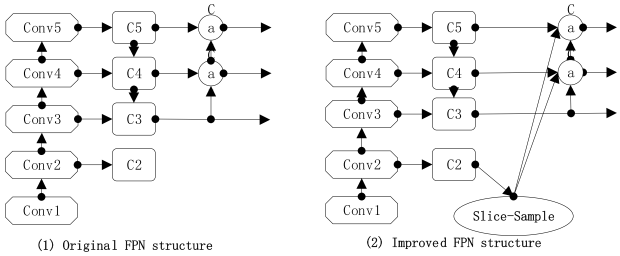
In order to improve the real-time detection of traffic signs and enhance the detection capability of YOLOv7-tiny for small objects, this research paper presented a novel algorithm that focuses on detecting small traffic signs in a realistic environment. The algorithm, as shown in Figure 1, exploited an augmented path from the bottom to the top on FPN to effectively utilize the fine-grained features of the lower convolutional layer. Furthermore, the integration of rich local region features was achieved through a downsampling module we proposed. It greatly enhanceed the accuracy of detecting small targets. To effectively integrate multiscale local region features within the downsampling module, we proposed the B2 channel attention mechanism. Finally, we introduced the EIOU loss function to minimize the optimization error of the network. Figure 1 illustrates the architectural components of the algorithm, where Conv represents a convolutional layer and SiLu denotes the activation function. BN corresponds to Batch Normalization, while MP represents a max pooling structure. E-ELAN refers to a structure consisting of convolutional layers, and E-SPP is a structure comprised of max pooling and convolutional layers. The detailed explanations of Slice-Sample and B2 can be found in the subsequent sections.

A. IMPROVED YOLOv7-TINY FPN

There are semantic differences between the features in different layers of the feature pyramid. The shallow layer has a small number of feature channels and a large feature scale, but it contains less semantic information and only some edge information. On the other hand, the deep layer has a large number of feature channels and contains more semantic information.

When fusing two adjacent scales for features, the high-level feature map is downscaled by 1 × 1 convolution to make the number of channels the same as that of the lower feature map, and then up-sampled by a factor of two to make the scale consistent; while the bottom feature map is downsampled to scale down to the same as that of the upper feature map, and then downscaled by 1 × 1 convolution to make the number of channels consistent.

The feature maps obtained by a series of downsampling in the yolov7-tiny backbone network are {C2,C3,C4,C5}. We argued that the downsampling of the model by convolution would lead to the disappearance of small object information in the process of feature fusion. Due to the C2 layer has the least number of downsampling, which contains the most information about small objects, so we improve the original FPN by downsampling the C2 layer and splicing it with C5 and C4 layers, so that more identifiable small object features could be incorporated into the training and detection. The FPN structure of the original model and the improved FPN structure are shown in Figure 2.



(1) Original FPN structure
(2) Improved FPN structure

FIGURE 2. Original FPN structure and improved FPN structure. Compared to the original FPN, we have allowed the bottom layer to merge with the top layer.

To optimize the extraction of information from the C2 feature map and reduce the number of parameters, we employed three downsampling methods. The first method was convolution. The second method was max pooling. However, the first method had the drawback of potentially causing the disappearance of small object features. The second method was not suitable for effective network learning due to the reduced parameter count. Consequently, neither of these approaches was suitable for our model. The schematic of convolution and max pooling are shown in Figure 3.

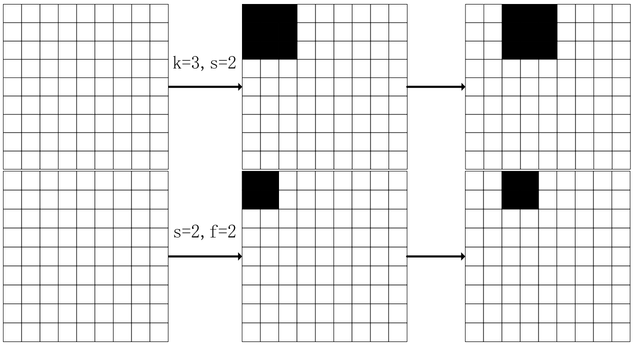


FIGURE 3. Schematic of convolution and max pooling. The above one is that we use convolution with k=3 and s=2. The another one is that we pool by s=2 and f=2.

The third method we use is Slice-sample, as shown in Figure. 4. For a $C \times W \times H\,(W = H)$ feature map, we divided it into C $W \times H$ feature maps according to the number of channels, and each W×H feature map is partitioned into 4 $W/2 \times H/2$ subgraphs according to the pixel-by-pixel taking form. Then, we loaded each subgraph into four different sets in order, so that there are C subgraphs size of $W{\times}H$ in each set. After that, each subgraph is taken out step by step in the order of the sets, and they are stitched together to form a feature map size of $4C \times W/2 \times H/2$. This completed the scale reduction. We then used the channel attention mechanism to assign weights to the generated feature maps to get the final feature map. Finally we reduced the number of channels by 1 × 1 convolution to allow the feature map to be spliced with other feature maps. In addition, since the pixels of small object feature occupy less of the whole map feature map. After the feature are separated and merged, the information between the channels is highly correlated, so we propose the B2 channel attention mechanism to enable the network to capture these correlations. The B2 channel attention mechanism is better suited for Slice-sample than the other channel attention mechanisms. We will discuss this in subsection B.

The advantage of this method was to ensure that the information of C2 feature map wasn't lost. We don't use convolution or average pooling to fuse the information of the features, nor do we use maximum pooling to select the information of the feature maps. This is inconsistent with the downsampling method proposed by many scholars. To explore the superior performance of Slice-sample, we compare it with some excellent downsampling methods such as Atrous Conv [42], Depthwise Separable Conv [43] in the experimental section.

In the FPN of YOLOv7-tiny network, the downsampling process from C5 to C4 and from C4 to C3 is achieved by applying convolution with kernel size (k) of 3 and stride (s) of 2. When we replaced this process with the Slice-Sample, we observed an increase in accuracy along with a decrease in the number of parameters.

B. B2 (TWO BRANCH) CHANNEL ATTENTION MECHANISM

In the field of neural networks, it is common to incorporate an attention mechanism, which is an additional neural network that can effectively select specific segments or assign varying weights to different parts of the input. The purpose of this attention mechanism is to extract and prioritize significant

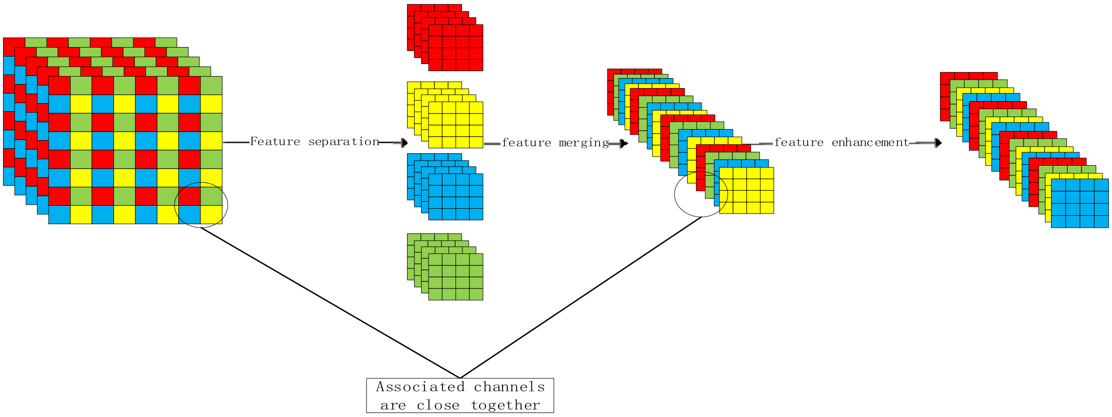


FIGURE 4. Schematic of slice-sample.

information from a vast quantity of data. Specifically, the channel attention mechanism operates by employing automatic learning within the channel dimension. It accomplishes this by leveraging a additional neural network that evaluates the importance of each channel in the feature map. Consequently, the resulting importance values are utilized to assign weight values to each feature, thereby enabling the neural network to concentrate on particular feature channels. By boosting the channels of the feature map that are pertinent to the given task and suppressing those that are less relevant, the network can optimize its performance.

The SE [30] channel attention is a classical channel attention module that follows a specific procedure. It begins with global averaging pooling of the input feature map,as shown in formula(1). Then, it goes through downscaling using a fully connected layer with the ReLU activation function. After that, it goes through upscaling using another fully connected layer with the sigmoid activation function. This process adjusts the channel weights to select more valuable channel feature information,as shown in formula(2).

$$gc = Asq(Fc) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} xc(i,j) \tag{1}$$

$$Zc = \rho(\delta(gc)) \tag{2}$$

where $X_c(i,j)$ denotes the value of the c channel at position $c(i,j)$; A_{sq} is the global average pooling function; F_c represents the input feature map with a size of $W \times H \times C$, while g_c refers to the attention matrix after global average pooling, with a size of $1 \times 1 \times C$.The sigmoid activation function is denoted by ρ, and the ReLU activation function is denoted by δ. The final output is given by formula (3).

$$Fc1 = Zc \cdot Xc \tag{3}$$

The use of a fully connected layer to obtain a global perceptual field does not allow the model to focus on the correlation between adjacent channels. Additionally, the fact that a fully connected layer increases the number of parameters in the network contradicts the construction of a small model.

Inspired by the ECA [31] attention mechanism, we proposed the B2 channel attention mechanism. First, the input feature map is pooled globally on average and globally on maximum, the formulas are shown in formula (4) and formula (5). Then, the feature channels are computed using a one-dimensional convolution with a convolution kernel size of 3. Finally, the output feature map is formed by multiplying the input feature map with the corresponding normalized weights on a channel-by-channel basis. The output feature map can be acquired using formula (6).

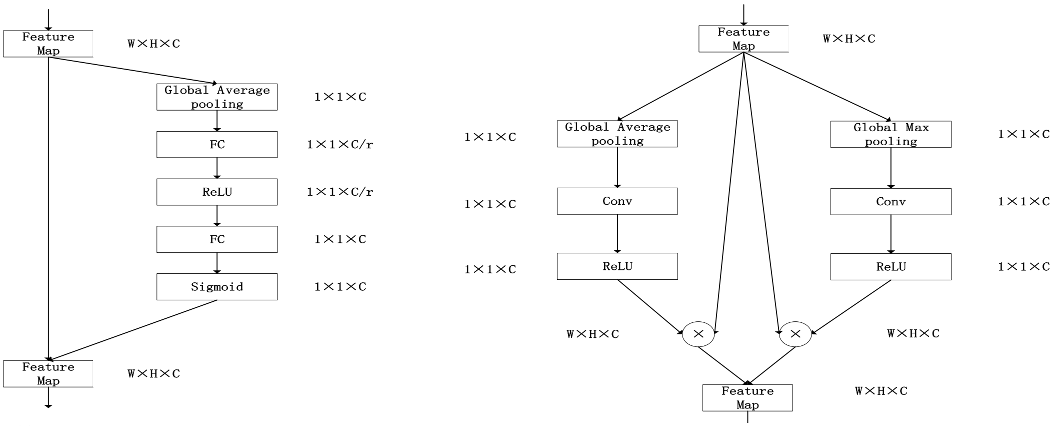
$$gc = Asq(Fc) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} xc(i,j) \tag{4}$$

$$hc = A\max(Fc) = \max \sum_{i=1}^{H} \sum_{j=1}^{W} xc(i,j) \tag{5}$$

$$Fc2 = \sigma(Conv(gc)) \cdot Fc + \sigma(Conv(hc)) \cdot Fc \tag{6}$$

where, σ is the ReLU activation function, F_c is the input feature map, g_c is the attention matrix after global average pooling, h_c is the attention matrix after global maximum pooling, and we chose a one-dimensional convolution with a convolution kernel of 3 to correlate between adjacent channels. The weights obtained, which have been normalized, are subsequently applied to the features of each channel. Figure 5 displays the structure of the B2 attention mechanism as well as the SE attention mechanism.

Compared to the SE channel attention mechanism, we use two branches to obtain the information of the input feature map, which allows more information to be extracted. The method using small convolution reduces the number of parameters of the model than the method using fully connected layers. More importantly, we can conclude from Figure 4 that the information of the feature map after slicing is dispersed among the neighboring subgraphs. Therefore, the B2 attention mechanism using small convolution can capture the correlation between neighboring channels well. As part of



(1)SEChannel attention mechanism

(2)B2 Channel attention mechanism

FIGURE 5. Compare with SE, B2 channel attention module has two branches that allow for more comprehensive information about small objects, while we use convolution to obtain the information between the channel can make the number of model parameters smaller.

the sliced sample, it is more suitable for subgraphs for feature enhancement.

From the experiment we can find that the B2 module has good performance not only combined with the Slice-Sample module, but also combined with the E-spp module at the backbone.

C. LOSS FUNCTION IMPROVEMENT

The loss function in YOLOv7-tiny is the CIoU [32] loss function with the following formula(7), formula(8) and formula(9). Where w, h, b, b_{gt} denote the width and height of the prediction box and the real box. b and b_{gt} denote the centroids of the prediction bounding box and the real bounding box. σ denotes the Euclidean distance of b and b_{gt}. w_c and h_c denote the width and height of the smallest outer rectangle of the prediction box and the true box, and IoU [33] denotes the intersection and merging ratio.

$$L_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \alpha\upsilon \tag{7}$$

Among them:

$$\alpha = \frac{\upsilon}{(1 - IOU) + \upsilon} \tag{8}$$

$$\upsilon = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{9}$$

Although CIoU increased the aspect ratios of the predictor and GT boxs, there were still the following problems: in the regression. When the aspect ratios of the two boxs were linear, the penalty term loses its original function. Consequently, w and h values in the gradient of the predictor frame didn't work. It didn't effectively describe the regression objective and might lead to slow convergence and inaccurate regression.

To address this issue, Zhang et al. proposed a method called EIoU [34]. This method preserves the overlap loss and center distance loss of CIoU, but modifies the width-height loss and penalizes the prediction results of w and h directly during the penalty process. This adjustment allows for a better response to the width-height difference between the prediction frame and the object frame, resulting in faster network convergence and improved regression accuracy. The equation for the EIoU function is shown in formula (10).

$$\begin{aligned} L_{EIOU} &= L_{IOU} + L - dis + L_{asp} \\ &= 1 - IOU + \frac{\rho^2(b, b^{gt})}{(w^c)^2 + (h^c)^2} + \frac{\rho^2(w, w^{gt})}{(w^c)^2} \\ &\quad + \frac{\rho^2(h, h^{gt})}{(h^c)^2} \end{aligned} \tag{10}$$

where, L_{IoU}, L_{dis}, L_{asp} respectively denote the loss, distance loss and width-height loss. w, h, w_{gt}, h_{gt}, respectively denote the width and height of the prediction box and the true box. b and b_{gt} respectively denote the centroids of the prediction bounding box and the true bounding box, ρ denote the Euclidean distance between b and $b_{gt..}$ The dimensions of the prediction box and the real box are represented by w_c and h_c, which respectively refer to the width and height of the smallest outer rectangle. IoU denotes the intersection and merging ratio.

In order to explore the loss function that is more suitable for our model, we also compared it with GIoU [45] and DIoU [44]. The results are shown in the experimental section.

III. EXPERIMENTAL RESULTS AND ANALYSIS

Experimental conditions: processor is AMDEPYC770264-CoreProcessor, memory is 50G, and graphics card is NVIDIAA100, 40GB. The programming environment of this paper is Python3 9, Torch1.12.1. The training parameters are shown in the table 1.

A. DATASET

This paper utilizes two commonly used datasets in the field of traffic sign detection. The first dataset is the German Traffic Sign Detection Benchmark (GTSDB) [35], which consists of

TABLE 1. The model training parameters.

Parameters	Parameter Value
Learning Rate	0.01
Batch size	8
Image size/pixels	640×640
Training volume	300
Number of warm-ups	3

600 training images and 300 test images. These images have a resolution of 1360 × 800 and encompass traffic signs under different lighting conditions. It is a foreign publicly available dataset that has been widely used by many research teams over the years. The GTSDB dataset has a balanced number of samples per category and contains a large number of small objects.

Since GTSDB categorizes traffic signs into three groups (mandatory, danger, and prohibited signs), a detection scheme that performs perfectly on GTSDB may not necessarily perform well in multi-category scenario. Therefore, the Tsinghua-Tencent 100K (TT100K) dataset [36] has been introduced. This dataset serves as a benchmark for Chinese traffic sign data and includes a large number of Chinese traffic sign images. Each image in TT100K has a resolution of 2048 × 2048 and covers traffic signs under diverse lighting conditions. The TT100K training set consists of 6105 images, while the test set contains 3071 images. The dataset consists of 221 categories, however, the distribution of samples across these categories is highly imbalanced. Some categories have a large number of samples while others have very few or even zero samples. To address this issue, we have chosen to focus on the 45 categories that have more than 50 samples for our experiments.

Overall, we chose GTSDB as the primary dataset for our experiment because it has a balanced number of samples per category and contains a large number of small objects. Many scholars have experimented on it. In addition, we conducted a supplementary experiment using the categories from TT100K that had a sample size greater than 50. This allowed us to evaluate the performance of our model in a multi-category scenario.

B. EVALUATION INDIATORS

In this paper, the differences in detection of several types of images by the improved network model before and after comparing the same experimental environment to assess the leakage and false detection, mainly by selecting the accuracy-recall (P-R) curve and the average accuracy (AP) and average precision mean (mAP), the size of model, where the formulae are as follows:

$$P = \frac{T_{TP}}{T_{TP} + F_{FP}} \times 100\% \tag{11}$$

$$R = \frac{T - TP}{T_{TP} + F_{FN}} \times 100\% \tag{12}$$

$$AAP = \int_0^1 P(R)dR \tag{13}$$

where: T_{TP} denotes correct prediction; F_{FP} denotes wrong prediction, which includes the cases of putting false detection and missing detection; F_{FN} denotes the case of mistakenly detecting traffic sign objects as other categories; P is the accuracy rate, as shown in formula (11). R is the recall rate, as shown in formula (12). In the P-R curve, the area enclosed by the P-R curve and the coordinate axis is equal to the value of AP, as shown in formula (13). The mAP can be obtained by averaging the AP values of all categories. In general, the mAP was used to evaluate the detection performance of the object detection model.

C. EXPERIMENTAL RESULTS

In the improved FPN, we added a path from the bottom of the backbone to the FPN. In order to better integrate the small target information in the feature map of the underlying backbone into the FPN, we use three downsampling methods in the path. The performance of these three downsampling methods is shown in Table 2.

According to Table 2, we can see that pooling has lower performance than the original network although it does not increase the parameters compared to convolution. Downsampling in the form of convolution improves the performance of

TABLE 2. Performance of different sampling method.

Network Model(on GTSDB)	Map0. 5	Map0.5 : 0.95	Model Size (MB)
Original FPN	0.8599	0.6706	22.97
Our FPN + Conv	0.8679	0.954	23.87
Our FPN+Max Pooling	0.8318	0.6379	21.68
Our FPN + Slice-sample	**0.9039**	**0.7186**	22.13
Our FPN + Atrous Conv[42]	0.863	0.6532	22.69
Our FPN + Depthwise Separable Conv[43]	0.8392	0.6402	23.17

TABLE 3. The effect of slice-sample in four different cases.

Network Model(on GTSDB)	Map0. 5	Map0.5 : 0.95
Model 1	0.861	0.6718
Model 2	0.9039	0.7122
Model 3	**0.9189**	**0.7346**
Model 4	0.873	0.6532

TABLE 4. Performance of B2 attention mechanisms and se attention mechanisms.

Conv1	Conv2	Conv3	Conv4	Conv5	Map0. 5	Map0. 5 : 0. 95
+B2					0.8712	0.6816
	+B2				0.8732	0.6795
		+B2			0.8781	0.6824
			+B2		0.8698	0.671
				+B2	**0.8866**	**0.705**
+SE					0.8534	0.6692
	+SE				0.8434	0.6695
		+SE			0.8672	0.6713
			+SE		0.8231	0.6991
				+SE	0.8712	0.6556

TABLE 5. Comparison of the original yolov7 family of algorithms with the yolov7 family of algorithms after adding all our methods.

Model	mAp0. 5	mAp0. 5 : 0. 95	Model Size (MB)
Yolov7-tiny	0.8599	0.6706	22.97
Yolov7-tiny + all our methods	**0.9347**	**0.7492**	23.29
Yolov7	0.9355	0.7657	141.96
Yolov7+all our methods	**0.9539**	**0.8092**	143.94
Yolov7-E6	0.936	0.778	206.01
Yolov7-E6 +all our methods	**0.941**	**0.797**	208.88
Yolov7x	0.9516	0.8276	270.21
Yolov7x + all our methods	**0.9572**	**0.8302**	273.98

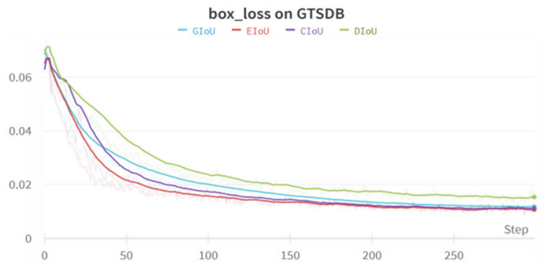


FIGURE 6. Visualization of box loss on GTSDB.

the network, but the number of parameters is also increased. Using the Slice-sample downsampling method not only reduces the number of parameters but also improves network performance. Furthermore, we conducted a comparison with other forms of convolution for downsampling, and the results demonstrate that Slice-sample outperforms them.

In order to investigate where the Slice-Sample module is best placed, we conducted the following experiments.

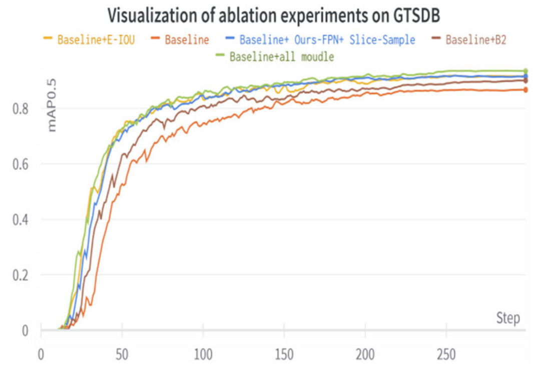


FIGURE 7. Visualization of ablation experiments on GTSDB.

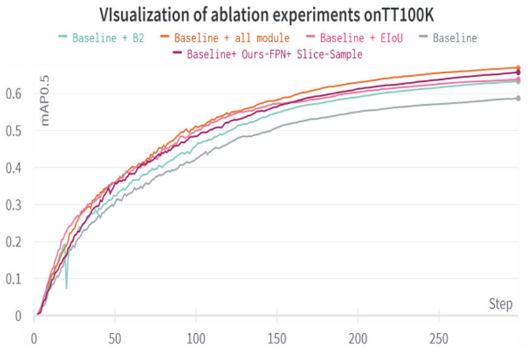


FIGURE 8. Visualization of ablation experiments on TT100K.

(1)Add Slice-Sample module in backbone(Model 1)

(2)Add Slice-Sample module in FPN(Model 2)

(3)Add Slice-Sample module in our proposed FPN (Model 3)

(4)Add Slice-ample module in all down-sampling modules in the network(Model 4)

According to Table 3, the inclusion of the Slice-Sample module in our proposed FPN yields better results and provides more valid information compared to the original network. This indicates that the improved FPN enables the model to learn more about the underlying small targets. Since adding

TABLE 6. Reslut of each module on GTSDB dataset.

Baseline(GTSDB)	Ours-FPN+ Slice-Sample	B2 Attentional Mechanism	E-IOU	mAp0. 5	mAp0. 5 : 0. 95	Model Size (MB)
√				0.8599	0.6706	22.97
√	√			0.9189	0.7346	23.2
√		√		0.8866	0.705	23.07
√			√	0.8941	0.693	22.97
√	√	√	√	**0.9347**	**0.7492**	23.29

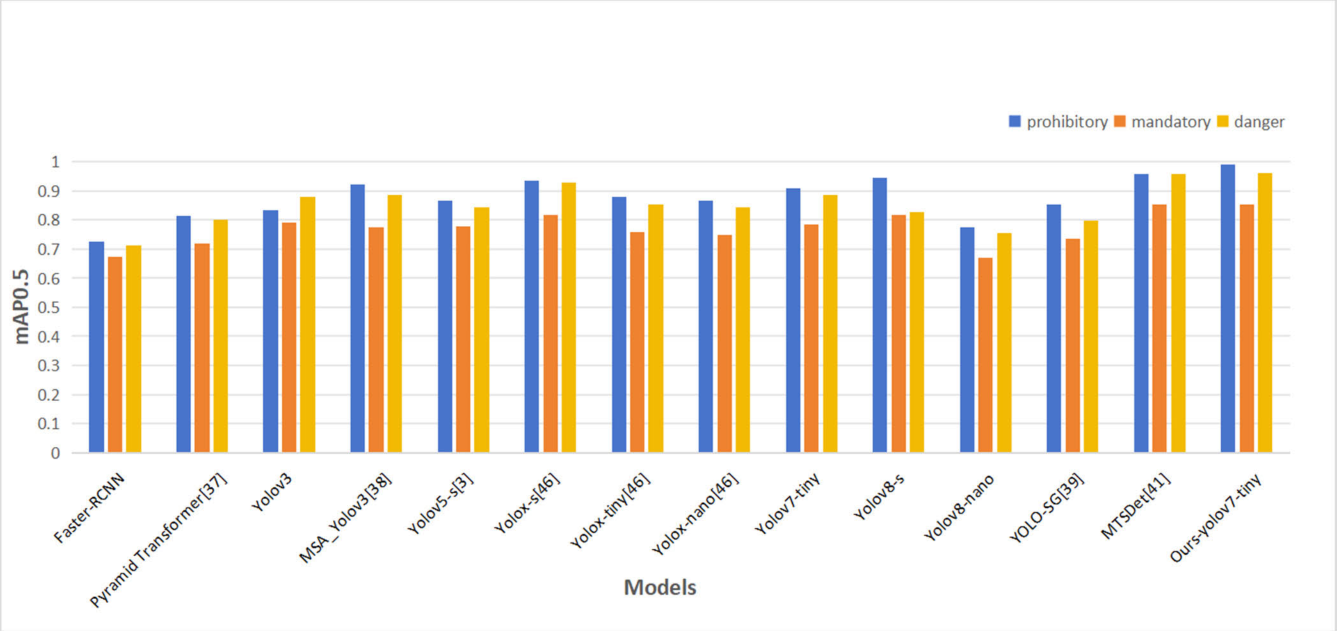


FIGURE 9. Performance of different networks on each category of the GTSDB.

the Slice-Sample module to the backbone network did not yield satisfactory results, we suggest that the downsampling module in the backbone network should use convolution and pooling for feature fusion.

In order to investigate the performance of the B2 attention mechanism and the SE attention mechanism, and determine the ideal layer in the backbone where they work best, we conducted a series of experiments (refer to Table 4). The results from the table indicate that both B2 attention mechanisms exhibit superior performance when combined with the backbone network compared to the SE attention mechanism. Furthermore, the B2 attention mechanism demonstrates optimal performance when integrated with the last layer of the backbone network.

In order to explore the effect of the loss function on the model, we compare the original loss function–CIoU with EIoU, GIoU [45], and DIOU [44], as shown in Figure 6. According to the experimental results, EIoU can allow the model to locate the object box faster. Before 50 steps, the box loss of EIoU is lower than that of CIoU, which means that the model using EIoU learns the location information faster than the model using CIoU. After 150 steps, the model with EIoU and the model with CIoU gradually leveled off in terms of box loss, with the former having a slightly lower box loss than the latter. In terms of box loss during the whole training process, EIoU performs better than the other three loss functions. CIoU has higher box loss than GIoU before 50 steps, but after 50 steps, CIoU performs better. DIoU performs lower than the other three loss functions throughout the training process.

Yolov7-tiny is one of the Yolov7 family of algorithms. In order to verify the generality of our method, we performed a comparison on the GTSDB dataset. According to the table 5, we can conclude that our proposed method improves among all the algorithms in the Yolov7 family. And the yolov7-tiny has the most performance improvement after adding our improvement methods.

Ablation experiments were conducted using YOLO7-tiny as the baseline model to demonstrate the contributions of each improvement in this paper to the model's

TABLE 7. Result of each module on TT100K dataset.

Baseline(TT100K)	Ours-FPN+ Slice-Sample	B2 Attentional Mechanism	E-IOU	mAp0. 5	mAp0. 5 : 0. 95	Model Size (MB)
√				0.587	0.4135	23.4
√	√			0.6534	0.411	23.65
√		√		0.638	0.4596	23.5
√			√	0.6342	0.4525	23.4
√	√	√	√	**0.6629**	**0.4729**	23.68

TABLE 8. Performance of different models on GTSDB dataset.

Models	Input Size	mAp0. 5	mAp0. 5 : 0. 95	Model Size (MB)	FPS
Faster-RCNN	640×640	0.703	0.5653	44	--
Pyramid Transformer[37]	640×640	0.778	--	74	--
Yolov3	640×640	0.8456	0.6529	--	37.59
MSA_Yolov3[38]	640×640	0.86	--	--	23.81
Yolov5-s[3]	640 × 640	0.8289	0.6477	27.9	52.9
Yolox-s[46]	640×640	0.893	0.7169	36.2	73
Yolox-tiny[46]	640×640	0.8302	0.6523	24.27	59.3
Yolox-nano[46]	640×640	0.819	0.648	22.19	57.4
Yolov7-tiny	640×640	0.8599	0.6706	22.97	60.2
Yolov8-s	640×640	0.8826	0.7019	32.95	74.1
Yolov8-nano	640×640	0.7325	0.6297	12.6	46.91
YOLO-SG[39]	640 × 640	0.796	--	4.0	131.6
FSADD[40]	640 × 640	--	0.739	--	--
MTSDet[41]	640 × 640	0.923	-	48.8	--
Ours-yolov7-tiny	640×640	**0.9347**	**0.7492**	23.29	67.5

performance. The evaluation was based on the parameters of the model and mAP. Tables 6 and 7 show that each module of the model improvement resulted in performance improvement on both the TT100K dataset and the GTSDB dataset. We used the TT100K dataset for additional experiments to validate the generalization of our methods. According to the Table 6, the addition of the Slice-sample downsampling module to our improved FPN led to a 5.9%(mAP0.5) improvement compared to the original model, highlighting the excellent performance of our downsampling module. Furthermore, the B2 attentional mechanism, in combination with the backbone, increased the mAP(0.5) by 2.67% compared to the original model, demonstrating its important role as a Slice-sample module and its effectiveness when used alone. Additionally, replacing the CIOU loss function with the EIOU loss function also improved the model's accuracy. When all the improved modules were combined, the model's performance significantly improved, resulting in a 7.48%(mAP 0.5) higher than the original model.

According to the Table 6, the improved model has an increase of 0.32MB size compared to the original model, which is attributed to the addition of the B2 attention mechanism. Although the B2 attention mechanism increases the size of model by a small amount, it provides a much improvement in model performance. The visualization of the ablation experiment on the GTSDB dataset is shown in Figure 7. The visualization of the ablation experiment on the TT100K dataset is shown in Figure 8.

In order to verify the effectiveness of the improved YOLOv7-tiny on GTSDB data, we conducted experiments comparing it with other network models. The size of model for most of the selected models is roughly similar to yolov7-tiny. These experiments were conducted under the same configuration environment and initial training parameters. The results, as shown in Table 8, demonstrate that our improved model outperforms other models of the same magnitude. The performance of the selected networks on each category of the GTSDB dataset is shown in Figure 9. All the networks perform best on the category of PROHIBITORY

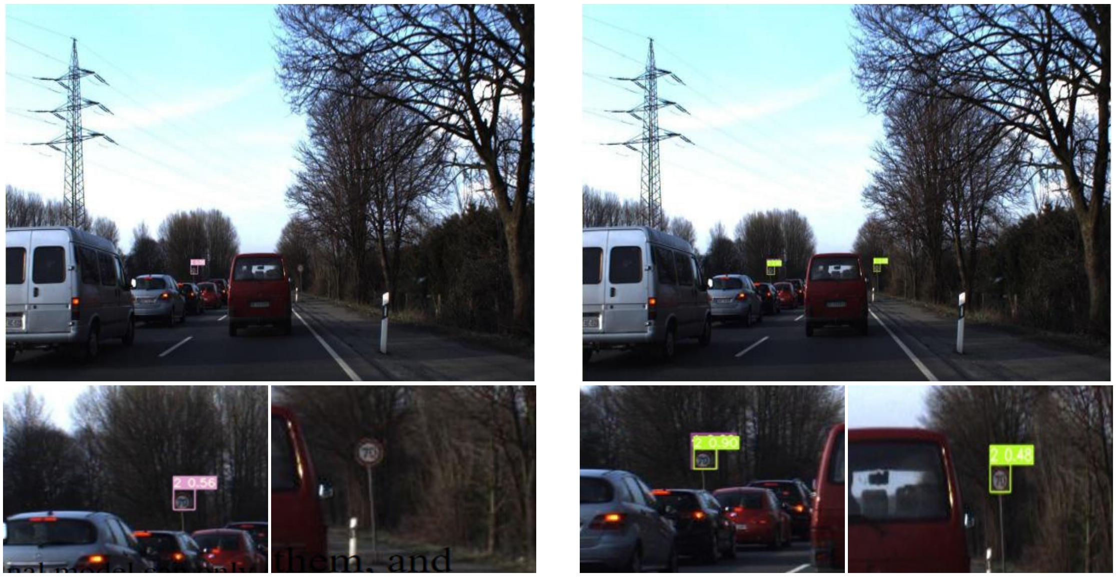


FIGURE 10. Comparison of small object detection results.

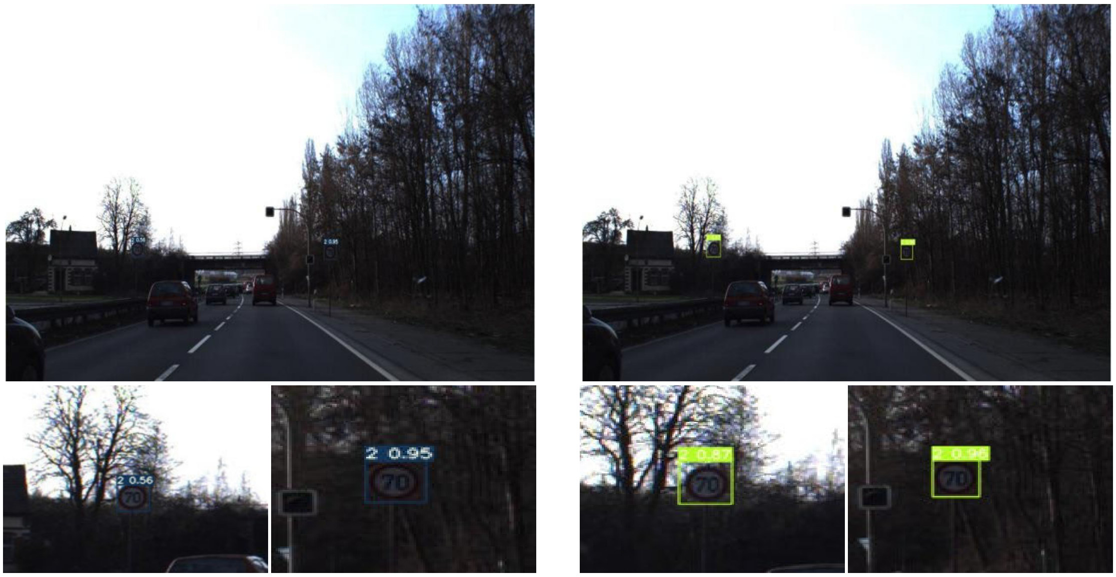


FIGURE 11. Comparison of detection results in a dim environment.

and poorly on the category of MANDATORY. Our improved yolov7-tiny achieves 99% detection accuracy on the category of PROHIBITORY, which is much better than the other networks.

D. ALIDATION OF PREDICTION EFFECT

In the GTSDB dataset, we selected images for detection in three cases. The first case is shown in Figure 10: in the very small object image with two traffic signs, the original model can only detect one of them, and its accuracy is only 56%, and the improved model can detect all the traffic signs, and its accuracy is 90%. The improved model is more capable of identifying small objects and has a larger accuracy improvement.

The second case is shown in Figure 11: there are two traffic signs under dim conditions, and the original model can detect both signs with an accuracy of 56% and 95%, respectively. The improved model can detect all the traffic signs and achieve 87% and 96% accuracy. The improved model has improved accuracy and is more capable of identifying dimmer objects.

The third case is shown in the figure 12: there are four traffic signs under the multi-object environment and the original model can detect all of them with the accuracy of 97%, 98%, 97% and 93%, respectively. The improved model can detect all the traffic signs with 98%, 94%, 97% and 99% accuracy. In the case of multiple objects and clear objects, the accuracy of the improved model is comparable to that of the Pre-improvement model.

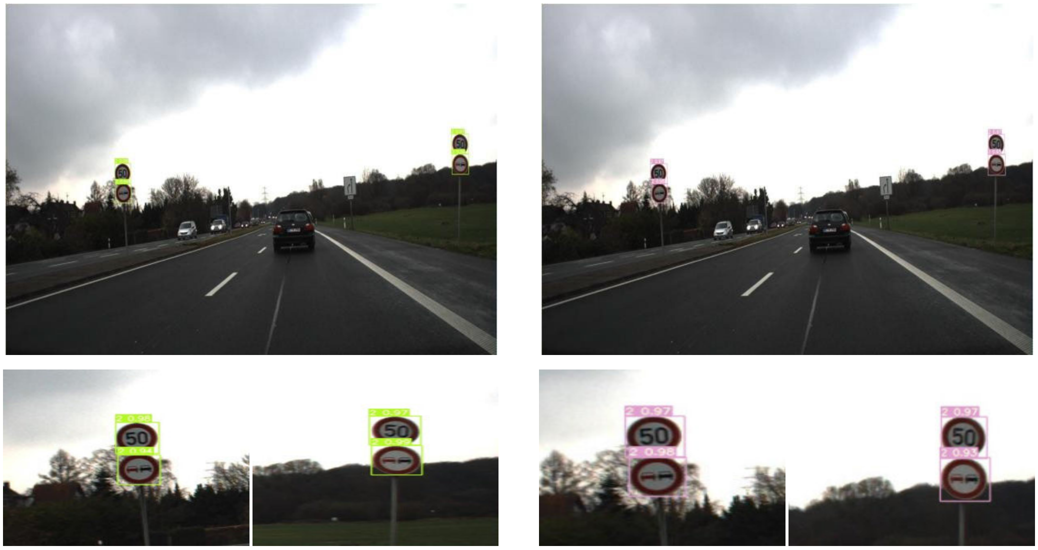


FIGURE 12. Comparison of detection results in a multi-object environment.

IV. CONCLUSION

In the traffic environment, we proposed an improved yolov7-tiny detection model for the difficulty of small object detection and the real-time of traffic signs. Based on the distribution of convolutional neural network features, a new FPN is proposed to integrate features rich in small object information into other features; through the idea of feature separation and merging, a Slice-Sample module is proposed for small object detection, and an B2 attention mechanism module is added. In addition, we propose a Slice-Sample module for small object detection, and incorporate the AMC attention mechanism module.

Through our experiments we found that the Slice-sample downsampling module works best when combined with our improved FPN. the B2 attention mechanism, which is part of the Slice-sample module, also performs well when combined with the backbone network alone. The EIOU loss function we introduced also performs well compared to the original loss function. Overall, the improved network shows good improvement over yolov7-tiny on both GTSDB and TT100K datasets. The improved yolov7-tiny is also the best compared to other models with comparable number of parameters.

The experiment has currently been conducted and validated on the GTSDB and TT100K datasets. Other aspects of detectors in traffic scenarios will be considered in the future. In addition, Our future focus will be on evaluating the performance of traffic sign detection at night and in bad weather conditions.

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] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[3] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788.

[4] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[9] Z. Li and F. Zhou, "FSSD: Feature Fusion Single Shot Multibox Detector," 2017, *arXiv:1712.00960*.

[10] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[12] X. Wang, A. Shrivastava, and A. Gupta, "A-Fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3039–3048.

[13] Z. He and L. Zhang, "Multi-adversarial faster-RCNN for unrestricted object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6667–6676.

[14] D. Tabernik and D. Skocaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1427–1440, Apr. 2020.

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

[16] A. Ruta, Y. Li, and X. Liu, "Towards real-time traffic sign recognition by class-specific discriminative features," in *Proc. Brit. Mach. Vis. Conf.*, 2007, pp. 1–10.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[18] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, ''Perceptual generative adversarial networks for small object detection,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1951–1959.

[19] Y. Rehman, J. Ahmed Khan, and H. Shin, ''Efficient coarser-to-fine holistic traffic sign detection for occlusion handling,'' *IET Image Process.*, vol. 12, no. 12, pp. 2229–2237, Dec. 2018.

[20] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, ''Traffic sign detection and recognition using fully convolutional network guided proposals,'' *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.

[21] E. Shelhamer, J. Long, and T. Darrell, ''Fully convolutional networks for semantic segmentation,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[22] J. Zhang, W. Wang, C. Lu, J. Wang, and A. K. Sangaiah, ''Lightweight deep network for traffic sign classification,'' *Ann. Telecommun.*, vol. 75, nos. 7–8, pp. 369–379, Aug. 2020.

[23] 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.

[24] Y. Cai, T. Luan, H. Gao, H. Wang, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, ''YOLOv4-5D: An effective and efficient object detector for autonomous driving,'' *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–13, 2021.

[25] 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.

[26] Y. Tian, J. Gelernter, X. Wang, J. Li, and Y. Yu, ''Traffic sign detection using a multi-scale recurrent attention network,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4466–4475, Dec. 2019.

[27] Y. Yuan, Z. Xiong, and Q. Wang, ''VSSA-NET: Vertical spatial sequence attention network for traffic sign detection,'' *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3423–3434, Jul. 2019.

[28] P. Liu, Z. Xie, and T. Li, ''UCN-YOLOv5: Traffic sign object detection algorithm based on deep learning,'' *IEEE Access*, vol. 11, pp. 110039–110050, 2023.

[29] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, ''YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors,'' 2022, *arXiv:2207.02696*.

[30] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, ''Squeeze-and-excitation networks,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[31] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, ''ECA-Net: Efficient channel attention for deep convolutional neural networks,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[32] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, ''Enhancing geometric factors in model learning and inference for object detection and instance segmentation,'' *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8574–8586, Aug. 2022.

[33] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, ''UnitBox: An advanced object detection network,'' in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 1–5.

[34] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, ''Focal and efficient IOU loss for accurate bounding box regression,'' *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.

[35] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, ''The German traffic sign recognition benchmark: A multi-class classification competition,'' in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2011, pp. 1453–1460.

[36] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, ''Traffic-sign detection and classification in the wild,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2110–2118.

[37] O. N. Manzari, A. Boudesh, and S. B. Shokouhi, ''Pyramid transformer for traffic sign detection,'' in *Proc. 12th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Nov. 2022, pp. 112–116.

[38] 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.

[39] Y. Han et al., ''YOLO-SG: Small traffic signs detection method in complex scene,'' *J. Supercomput.*, 2023, doi: 10.1007/s11227-023-05547-y.

[40] J. Chung, S. Park, D. Pae, H. Choi, and M. Lim, ''Feature-selection-based attentional-deconvolution detector for German traffic sign detection benchmark,'' *Electronics*, vol. 12, no. 3, p. 725, Feb. 2023.

[41] H. Wei, Q. Zhang, Y. Qian, Z. Xu, and J. Han, ''MTSDet: Multi-scale traffic sign detection with attention and path aggregation,'' *Int. J. Speech Technol.*, vol. 53, no. 1, pp. 238–250, Jan. 2023.

[42] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, ''Deformable convolutional networks,'' in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.

[43] F. Chollet, ''Xception: Deep learning with depthwise separable convolutions,'' in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.

[44] X. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, ''Distance-IoU loss: Faster and better learning for bounding box regression,'' in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 12993–13000.

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

[46] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, ''YOLOX: Exceeding YOLO series in 2021,'' 2021, *arXiv:2107.08430*.

FEIFAN SHE was born in Hubei, China, in August 1998. He received the bachelor's degree in traffic engineering from Wuyi University, in 2021, where he is currently pursuing the master's degree. His research interest includes traffic detection engineering.

ZHIYONG HONG received the B.S. and M.S. degrees in computer science and technology from the Shenyang Institute of Technology, Shenyang, China, in 2001 and 2004, respectively, and the Ph.D. degree in computer science and technology from Southwest Jiaotong University, Chengdu, China, in 2014. He is a Professor with the Facility of Intelligence Manufacture, Wuyi University. His research interests include intelligent information processing, AI, block chain, and rough set theory.

ZHIQIANG ZENG was born in Shaoguan, Guangdong, China, in 1989. He received the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2018. Since then, he has been a Lecturer with the Faculty of Intelligent Manufacturing, Wuyi University, China. He is the author of a number of high-level articles. His research interests include machine learning, data mining, industrial big data, and production scheduling.

WENHUA YU is currently a Visiting Professor with Wuyi University, Jiangmen, China. He has published more than 200 technical aritcles and ten books on the topics related to AI, machine learning, computational electromagnetics, and parallel processing. He served as a member for technical committees and international advisor committees of international conferences. He was selected as a ACES Fellow in 2018. He served as the general chair for several international conferences.