RBL-YOLOv8: A Lightweight Multi-Scale Detection and Recognition Method for Traffic Signs

Shijie Guo, Nannan Zhao, Xinyu Ouyang, Yifan Ouyang

Abstract—**To address the problems of misdetection, omission, and low accuracy in traffic sign detection and recognition, a novel method called RBL-YOLOv8 is presented by improving YOLOv8. In the feature extraction network, the RepNC-SPELAN module is used to replace the C2f module to improve the feature extraction capability and reduce the number of parameters. In the feature fusion network, fusion of large-scale feature layers is added, while weighted feature fusion is used to create cross-layer connections between shallow and deep features to improve the utilisation of shallow features for better detection of small targets. A lightweight detection head is proposed to reduce the number of parameters and computational complexity of the model, while improving the localization and classification ability of the detection head. The MPDIoU loss function is used to replace CIOU, which can better accelerate the bounding box regression. The improved model is conducted experiments on the CCTSDB and TT100K datasets and compared with other algorithms, the results validate its effectiveness and superiority.**

Index Terms—**traffic sign detection, YOLOv8, multi-scale detection, shared convolution, lightweight.**

I. Introduction

TRAFFIC sign detection and recognition task is one of the important technical foundations of intelligent transportation systems and unmanned systems. This technology refers to the detection and recognition of traffic signs in images or videos through image processing and deep learning, which plays an important role in guiding the subsequent behavior of intelligent transportation systems and unmanned systems [1]. To improve traffic safety, it is necessary to continuously improve the accuracy and inference speed of traffic sign detection and recognition algorithms, so how to detect and recognize traffic signs quickly and accurately is an urgent problem in the field of object detection. On roads, traffic signs are usually distinguished from the environment by eye-catching colors (red, yellow, and blue) and specific shapes (triangles, circles, squares, and polygons) to convey warning, prohibition, and mandatory signals to vehicles and pedestrians to enhance the recognizability of traffic signs. Therefore, early traffic sign detection and recognition methods have been studied using the characteristics of traffic sign colors and shapes [2, 3]. Color-based methods use the RBG color space or HIS color space of color images to extract the features of traffic signs[4, 5], however, color-based methods are sensitive to light variations and are ineffective in detecting faded traffic signs. Shape-based detection methods manually extract features and set classifiers for traffic sign detection[6–8], however, shape-based methods are sensitive to scale changes and are ineffective in detecting deformed and occluded traffic signs, which makes it difficult for traditional traffic sign detection methods to be applied on real roads.

Deep Convolutional Neural Networks have been increasingly adopted in object detection due to their robust feature extraction capabilities [9, 10]. Advanced models such as the two-stage algorithms R-CNN [11], Fast-RCNN [12], and Faster R-CNN [13] have emerged. These two-stage algorithms first generate candidate regions, followed by candidate region classification and bounding box regression to achieve object detection. In contrast, one-stage algorithms such as YOLO [14], SSD [15], and SPP-Net [16] directly extract features from images using a single network for object detection and classification. Although two-stage algorithms have higher detection accuracy, they require more computational complexity and more labelled data. Compared to the one-stage algorithm, the two-stage algorithm has a slower detection speed and is not suitable for real-time detection scenarios. Recent research efforts have focused on improving deep learning models for traffic sign detection. For example, Wang et al. [17] reduced feature information loss by changing the attention module and feature enhancement module. Sun et al. [18] improved small object detection by expanding the detection scale and incorporating Coordinate Attention (CA) to facilitate rapid region localisation. Zhou et al. [19] extended the model's receptive field by combining depth-separable convolutions with different expansion rates to integrate contextual information and minimise information loss. Zhang et al. [20] improved spatial and positional focus by adding a Convolutional Block Attention Module (CBAM) to the YOLOv8 backbone network, reducing feature information loss during downsampling. Several studies have proposed novel techniques to improve traffic sign detection, such as using Swin Transformer modules [21], incorporating ResNeSt for feature extraction with coordinate attention mechanisms [22], and improving feature pyramids through

Manuscript received May 24, 2024; revised October 8, 2024. This work is supported by the Fundamental Research Funds for the Liaoning Universities of China (Grant No. LJ212410146005).

Shijie Guo is postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China. (e-mail: 1437940802@qq.com.)

Nan-Nan Zhao is professor of the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China. (Corresponding author, e-mail: 723306003@qq.com.)

Xinyu Ouyang is professor of the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, 114051, China. (Corresponding author, e-mail: 13392862@qq.com.)

Yifan Ouyang is undergraduate student of School of Electrical Engineering and Artificial Intelligence, Xiamen University Malaysia, Sepang, Selangor Darul Ehsan, 43900, Malaysia. (e-mail: AIT2209084@xmu.edu.my.)

path aggregation [23]. The above literatures have addressed many issues, however, there are still some challenges, such as the scale variation of traffic signs, light variation, and occlusion, which can affect the detection effect, and the need for lightweight models for deployment on resource-constrained devices.

Aiming at the above problems, the one-stage algorithm YOLOv8 is improved from four aspects, namely, feature extraction network, feature fusion network, detection head, and loss function, to improve the traffic sign detection model's ability to be applied in real scenarios. The improved method improves the detection accuracy, reduces the model size, has good real-time performance, and meets the lightweight requirement of mobile devices for deploying the model while improving the detection effect and robustness of the detector to multi-scale targets. The main four contributions are as follows:

(1) Improvement of the feature extraction network. The C2f module in YOLOv8's feature extraction network has been replaced by the lightweight feature extraction module RepNCSPELAN [24], and the feature extraction capability of the model is improved by using the network structure of RepNCSPELAN for efficient feature aggregation and the reparameterisation technique.

(2) Improvement of the feature fusion network. Increase the model's focus on the large-scale feature layer by fusing the P2 layer. Combining the idea of BiFPN [25], weighted feature fusion is used to fuse shallow and deep features of the same size to improve the utilisation of shallow features and improve the detection of multi-scale targets.

(3) Improvement of the detection head. The convolution in the detection head is improved using the GroupNorm layer [26] to improve the ability of the detection head to localize and classify. The number of 3*3 convolutions in the detection head is reduced using shared convolution [27], and the size of the bounding box is adjusted by feature scaling, which reduces parameters and computational complexity of the YOLOv8 decoupling head [28].

(4) Loss function replacement. The loss function CIoU [29] has been replaced with MPDIoU [30] to regress the bounding box by minimising the Euclidean distance of the diagonal line, which takes into account the loss factors such as overlapping area, distance from the center point, width, etc. of the predicted box and the ground truth box, enhancing the regression ability of the bounding box while accelerating convergence speed.

II. Related work

YOLOv8 is a one-stage target detection algorithm [31], and the entire network consists of the following four parts.

A. Input part

Scaling the input images to a fixed size ensures that all images have the same dimensions and can be batch-processed. Using the mosaic data enhancement technique, the images are randomly scaled, cropped, and aligned, which is used to increase the diversity of the samples and improve the robustness of the model. The pre-processed images are fed into the feature extraction network.

B. Feature extraction network

The Feature Extraction Network is a collection of high performance classifiers designed to extract feature information from targets within the image and consists mainly of the CBS and C2f modules. The CBS module plays a crucial role in downsampling the feature maps. As the primary feature extraction module, the C2f module contains the ELAN structure from YOLOv7 [32] and maximises the use of the bottleneck module to enhance gradient information. In particular, the C2f module excels at capturing richer gradient information with fewer parameters, contributing to a lighter model. Deep features are then extracted and fused using the SPPF module. During the fusion process, the maximum pooling module is iteratively applied to extract additional semantic detail from the deep features, and the resulting outputs are fed into the feature fusion network.

C. Feature fusion network

The feature fusion process of YOLOv8 is located between the feature extraction and the detection head, and the feature information at different levels of the feature extraction network is processed and fused and then passed to the detection head to further improve the multi-scale feature expression capability and robustness of the network. The feature fusion network utilizes the upsampling, CBS, and C2f modules to construct the FPN-PAN network structure [33, 34], which fuses the features of targets at different scales and improves the feature fusion capability of the network, thus improving the ability of the model to detect targets at different scales.

D. Detection head

The YOLOv8 uses decoupled headers in the form of bounding box regression and classification, respectively. Among them, the classification loss uses Binary Cross Entropy (BCE), while the bounding box regression uses a combination of Distributional Focal Loss(DFL) [35] and CIoU loss function. DFL enables the model to learn the loss of the position around the label, and together with the Anchor-Free centroid-based computation method (first determining the center region, then predicting the distance from the center to the four edges), it improves the model's performance in occlusion detection effect in case of occlusion. The allocation strategy for positive and negative samples is Task Aligned Assigner [36], which enables the model to continuously improve its sample allocation capability with training, and selects positive samples based on the weighting of classification and regression scores to complement the model's need for positive samples in different tasks.

III. Improved model

According to different application scenarios, YOLOv8 uses the scaling factor to divide the model into five versions, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x. According to the real-time and lightweight requirements of traffic sign detection and recognition tasks, YOLOv8n is selected as the benchmark model for experiments, and improvements are made in four aspects, namely, feature extraction network, feature fusion network, detection head, and loss function, respectively, and the improved method is called RBL-YOLOv8, and the overall network structure is shown in Fig. 1.

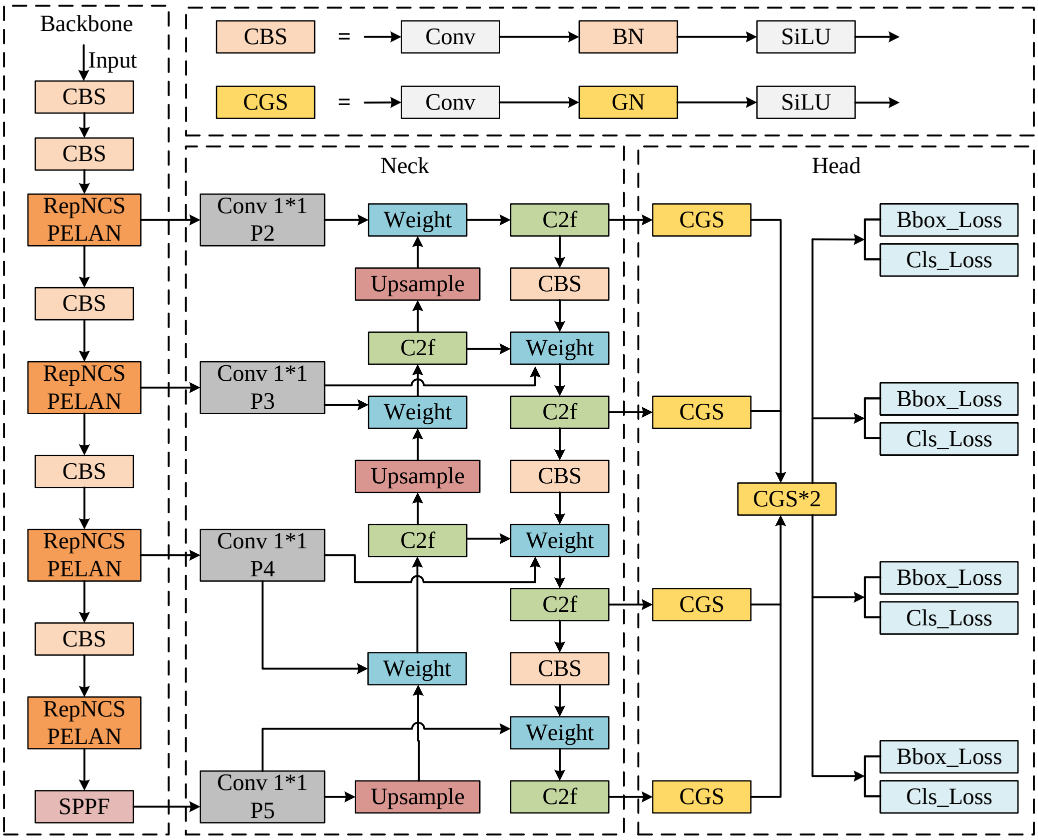


Fig. 1. The RBL-YOLOv8 structure diagram

A. Improved Feature Extraction Network

The role of the feature extraction network is to convert the input image into high-level feature representations that contain key information in the image, such as edges, texture, and shape. In YOLOv8, it uses CBS module for downsampling and then C2f module for feature extraction. In traffic sign detection tasks, the model is required to have high detection accuracy and lightweight model size for easy deployment in resource-constrained mobile and edge devices. Therefore, RepNCSPELAN is used instead of the C2f module as the new feature extraction module, aiming to achieve high accuracy while keeping the network lightweight. Thus, a Generalized Efficient Layer Aggregation Network (GELAN) was given by combining two gradient path planning neural network structures, CSPNet [37] and ELAN [38], as shown in Fig. 2(c).

The CSPNet divides the input feature map into x_1 and x_2, in the channel dimension, where the feature map x_1 is fused with the feature map x_2 after passing through a series of arbitrary feature extraction modules, as shown in Fig. 2(a). However, the feature map x_1 passes through several feature extraction modules resulting in the loss of some shallow feature information. In order to compensate for the lost information, GELAN combines ELAN based on CSPNet and fuses the feature map x_1 with the output of each feature extraction module it passes through with the feature map x_2, thus realizing a gradient structure with richer information.

Based on GELAN, RepNCSPEALN uses RepNCSP and CBS modules as feature extraction modules as shown in Fig. 3. The reparameterisation technique is used to increase the gradient feedback path by using multiple branches (e.g. multiple convolutional layers) in the training phase, and then reparameterize the parameters of each branch to the main branch in the inference phase. This reduces computation and memory consumption and improves the inference efficiency of the model. Taking the fusion of convolutional and BN layers as an example, the calculation of the convolutional layer is shown in Equation (1):

$$Conv\left(x\right) = W\left(x\right) + b \tag{1}$$

Where $W\left(x\right)$ is the weight and b is the bias, the BN layer is calculated as follows:

$$BN\left(x\right) = \gamma \times \frac{x - mean}{\sqrt{var}} + \beta \tag{2}$$

Here γ and β are parameters learned during training, representing the weights and biases of the BN layer, respectively, with mean representing the mean and var representing the variance. In the inference stage, the convolutional and BN layers are merged while the parameters are remapped as shown in Equation (3):

$$\begin{aligned} BN\left(Conv\left(x\right)\right) &= \gamma \times \frac{Conv\left(x\right) - mean}{\sqrt{var}} + \beta \\ &= \gamma \times \frac{W\left(x\right) + b - mean}{\sqrt{var}} + \beta \\ &= \frac{\gamma \times W\left(x\right)}{\sqrt{var}} + \left(\frac{\gamma \times \left(b - mean\right)}{\sqrt{var}} + \beta\right) \end{aligned} \tag{3}$$

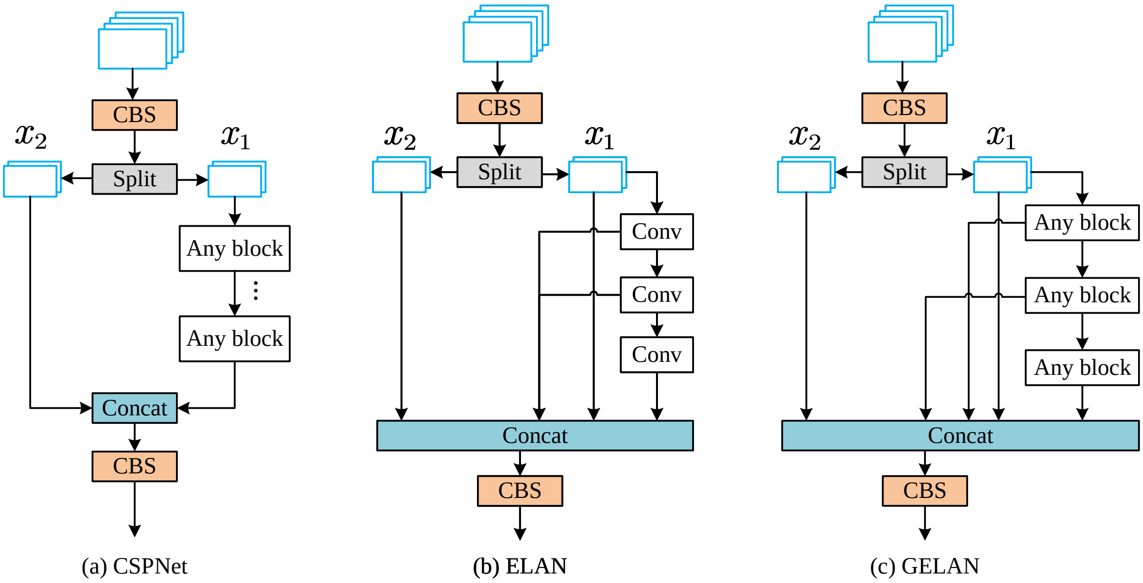


Fig. 2. The structures of CSPNet, ELAN and GELAN

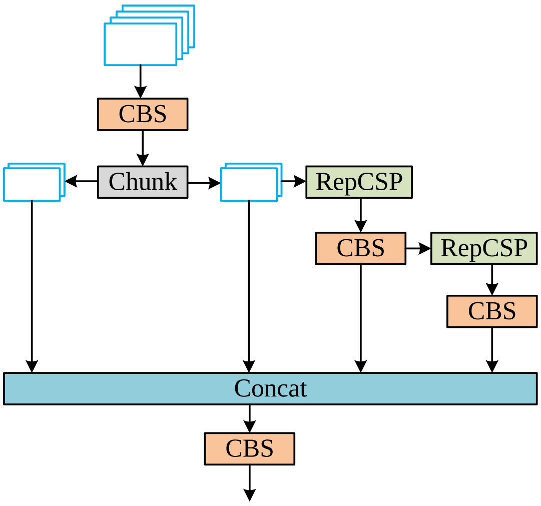


Fig. 3. RepNCPSELAN structure

Where $\frac{\gamma\times W(x)}{\sqrt{var}}$ is the fused weights and $\frac{\gamma\times(b-mean)}{\sqrt{var}}+\beta$ is the fused bias.

B. Improved Feature Fusion Network

YOLOv8 uses a feature extraction network to divide the feature map into five different scales, P1-P5. The FPN fuses P3 with B3, and P4 with B4, forming a top-down feature pyramid. However, the FPN structure results in the loss of some location information during the fusion process. To compensate for the positional information lost by FPN, YOLOv8 combines FPN and PAN to construct a top-down and bottom-up structure, as shown in Fig. 4. It builds a deeper network that retains more feature information than the FPN by fusing B4 with N4 and P5 with N5. Although the FPN-PAN structure fuses more feature layers and enriches semantic and location information, the FPN-PAN structure only fuses feature maps at the P3, P4 and P5 scales, and lacks attention to the large-scale feature P2, which leads to the loss of some valuable information, such as the color, texture, and shape information contained in the P2 feature map. In addition, the original features lose some of the feature information during up-sampling and down-sampling as the network depth deepens, resulting in lower feature information utilization, which limits the feature fusion capability and reduces the detection performance of the model, so the FPN-PAN structure needs to be further optimized.

The feature information of small-sized traffic signs is usually contained in the shallow feature map, and FPN-PAN has limited utilization of the shallow feature information. To solve this problem, the utilization of the shallow feature information is improved by reconstructing the feature fusion network. First, the P2, P3, P4 and P5 layers are each passed through a 1*1 convolutional layer to obtain the same number of channels, which can reduce the parameters in the feature fusion process. Then, the B2 layer is obtained by up-sampling the B3 layer, and the B2 layer is fused with the P2 layer to increase the scale of feature fusion, which can improve the attention to the large-scale feature layer. In addition, the idea of BiFPN is introduced to improve the utilization of feature information in layers P3 and P4, specifically, the feature map fused from B2 and P2 is downsampled to obtain N3, and an extra path is added between layers P3 and N3, P4 and N4 using cross-layer connection to fuse layers P3 and N3, P4 and N4, as shown in Fig. 5.

YOLOv8 fuses feature maps of different resolutions by summing them after resizing them to the same size, but the information contained in the feature maps of different resolutions does not contribute equally to the output feature maps. To solve this problem, a weighted feature fusion method is used to add additional learnable weights to each input feature map, so that it can automatically update the weights during the learning process, thus achieving a more reasonable feature fusion, the specific steps are as follows:

First, additional weights are added to each of the input feature maps I_i as shown in Equation (4):

$$O=\sum_i w_i\cdot I_i \tag{4}$$

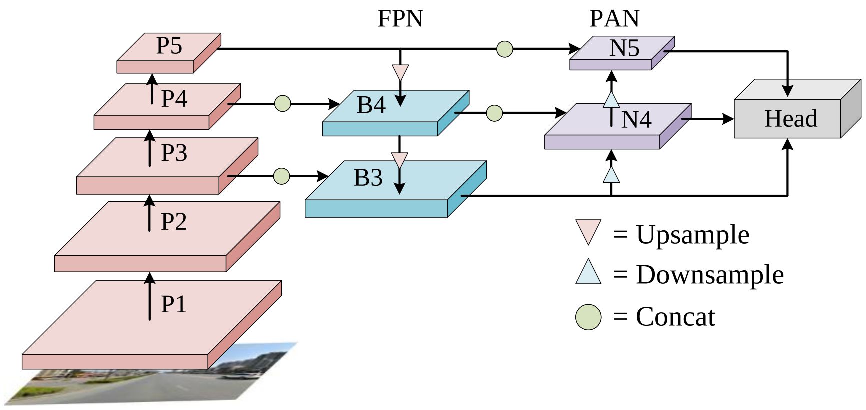


Fig. 4. The FPN-PAN structure in YOLOv8

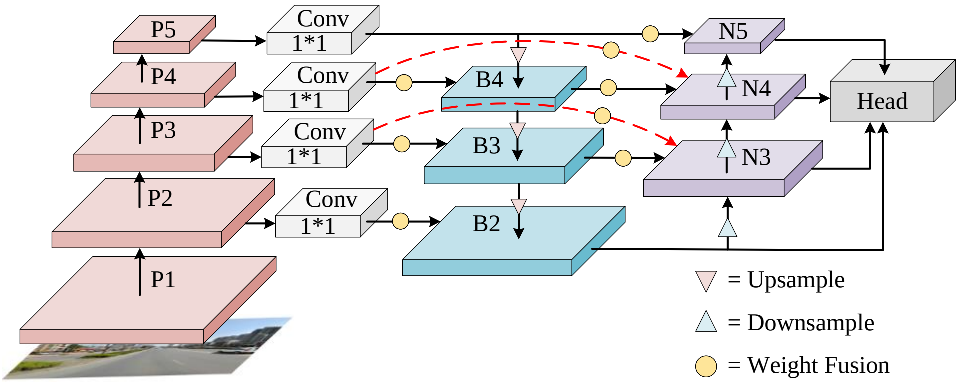


Fig. 5. The improved Feature Fusion Network structure

Where w_i is the learnable weight. If w_i corresponds to a scalar weight, since scalars are unbounded, leaving them unbounded will lead to an unstable training process, so a Softmax treatment is added for each weight, limiting the range of values for each weight to be between 0 and 1, as shown in Equation (5):

$$O = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}} \cdot I_i \tag{5}$$

However, by adding the exponential operation to the Softmax operation, the processing speed of the GPU is significantly reduced. To solve this problem and reduce the computational cost, the exponential operation in Eq. (5) is removed, and a ReLU function is added after each weight to ensure $w_i \geq 0$, and a minimum $\epsilon = 0.0001$ is added to ensure that the value is stable, as shown in Equation (6).

$$O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i \tag{6}$$

In this way, the value range of each normalized weight is still between 0 and 1, but the Softmax operation is avoided and the computational efficiency is improved.

C. Lightweight Detection Head

Many ITS systems use embedded devices, which usually have limited computing resources and storage space. The lightweight model reduces hardware requirements and is easier to deploy on cloud platforms or edge devices, improving the deployability and flexibility of ITS, which is crucial for large-scale deployment scenarios.

Through experiments, it is found that the number of parameters and computation of the YOLOv8 detection head occupies almost half of the whole model, which greatly increases the complexity of feature decoupling. Therefore, it is crucial to lightweight the detection head, which can effectively reduce the number of parameters and computational complexity of the model. The YOLOv8 detection head adopts a decoupled form, which divides the detection head of each scale into two branches, and outputs the localization loss and the classification loss after a series of convolutional operations, as shown in Fig. 6. In the convolution process, there are two 3*3 CBS modules and one 1*1 regular convolution, where Pi represents the feature maps of different scales. Thus each scale of the detection head contains four 3*3 convolutional layers, which is the main reason for increasing the number of detection head parameters.

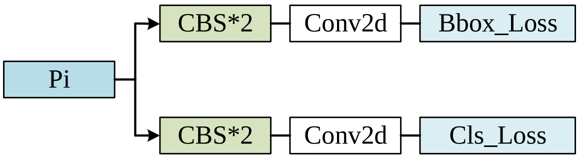


Fig. 6. YOLOv8 detection head structure

To optimize the detection head, a lightweight detection head scheme, LGSCD (Lightweight GroupNorm Share Conv Detection Head), is proposed, which aims to optimize the structure of the detection head of YOLOv8, and to reduce the parameters and the computational complexity of the model. LGSCD first utilizes 1*1 convolution to adjust the number of channels for different scales of feature maps to reduce the computational effort of the model. Then, a shared convolution consisting of two 3*3 convolutions is used to traverse the feature maps from different scales, and the output is divided into localization loss and classification loss. In addition, replacing BatchNorm with GroupNorm in the CBS module and improving CBS to CGS improves the localization and classification performance of the detection head[39]. The structure of LGSCD is shown in Fig. 7.

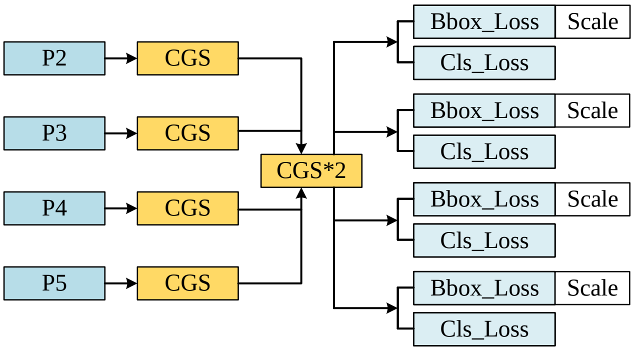


Fig. 7. LGSCD structure

For the input feature map $R^{C\times H\times W}$, the number of parameters after convolution is shown in equation (7):

$$Parameters = c_1 \times c_2 \times k^2 \tag{7}$$

Where c_1 denotes the number of channels of the input feature map, c_2 is the number of output channels after convolution, and k denotes the convolution kernel size.

Since the reconstructed feature fusion network has 4 scales of output, the number of parameters of the YOLOv8 detection head is shown in Equation (8):

$$\begin{aligned} &Parameters\left(YOLOv8\right) = c_1 \times c_2 \times 3^2 \times 16 \\ &= 144 \times c_1 \times c_2 \end{aligned} \tag{8}$$

The number of parameters of LGSCD is as follows:

$$\begin{aligned} &Parameters\left(LGSCD\right) = c_1 \times c_2 \times 3^2 \times 6 \\ &= 54 \times c_1 \times c_2 \end{aligned} \tag{9}$$

By comparing Eq. (8) and Eq. (9), the number of parameters in LGSCD has been reduced by more than half compared to YOLOv8 detection head, achieving lightweighting.

D. MPDIoU Loss Function

YOLOv8 uses CIoU Loss to calculate the bounding box loss. CIoU Loss introduces centroid distance loss and aspect ratio loss. The calculation formula is shown in Equation (10).

$$\begin{aligned} &L_{CIOU} = 1 - IOU\left(A,B\right) + \rho^2\left(A_{ctr},B_{ctr}\right)/c^2 + \alpha v \\ &v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \\ &\alpha = \frac{v}{(1-IOU)+v} \end{aligned} \tag{10}$$

Where ρ denotes the Euclidean distance, A_{ctr} and B_{ctr} represent the centroids of A and B, c denotes the diagonal length of the smallest bounding box enclosing A and B, and v and α represent the difference in aspect ratio. However, when the aspect ratios of the predicted box and the ground truth box are the same, the aspect ratio loss of the CIoU loss is constant at 0, resulting in no gradient backpropagation, which leads to the inability to continue learning the aspect ratio loss, which is detrimental to the bounding box regression. To solve this problem, the loss function MPDIoU is used to replace the CIoU in the YOLOv8, and the MPDIoU regresses the predicted box by minimizing the Euclidean distances between the upper-left and lower-right corners of the predicted box and the ground truth box, as shown in Fig. 8.

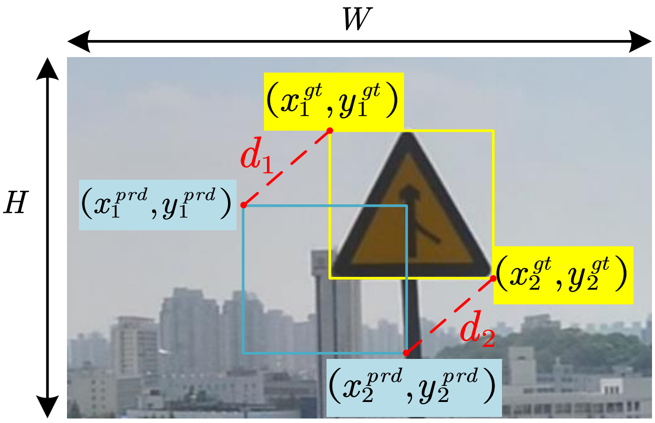


Fig. 8. MPDIoU regression method

MPDIoU Loss is calculated as shown in equations (11) through (14):

$$d_1^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2 \tag{11}$$

$$d_2^2 = \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2 \tag{12}$$

$$MPDIoU = IoU - \frac{d_1^2}{w^2+h^2} - \frac{d_2^2}{w^2+h^2} \tag{13}$$

$$L_{MPDIoU} = 1 - MPDIoU \tag{14}$$

Where w and h represent the width and height of the picture, respectively, and $\left(x_1^{gt}, y_1^{gt}\right)$ represent the coordinates of the upper-left and lower-right corners of the ground truth box, and $\left(x_2^{gt}, y_2^{gt}\right)$ represent the coordinates of the upper-left and lower-right corners of the predicted box, and $\left(x_1^{prd}, y_1^{prd}\right)$ and $\left(x_2^{prd}, y_2^{prd}\right)$ represent the Euclidean distance between the upper-left and lower-right corners of the predicted box and the ground truth box, respectively.

IV. Experiment

A. Experimental Environment and Parameter Setting

The input image size is 640*640, the initial value of the learning rate is 0.001, the momentum size is 0.98, the weight decay parameter is 0.001, the batch size is set to 32, the model is based on the PyTorch framework using CUDA 11.7, Windows system, the graphics card model is NVIDIA GeForce RTX 3060, the graphics card RAM size is 12G, memory size is 32G, processor model is 12th Gen Intel(R) Core(TM) i5-12490F.

B. Experiment Dataset

CCTSDB 2021 [40] is selected as the baseline dataset for the experiment, which includes images under six types of weather: foggy, snowy, rainy, evening, cloudy, and sunny, the extreme weather can better validate the robustness of the model. The dataset categorizes the signs into three categories: Mandatory, Prohibitory, and Warning as shown in Fig. 9. In order to improve the generalization ability of the model, 1004 images with a resolution of 2048*2048 are added to CCTSDB 2021, and the number of targets in the Warning category is increased by 1342 to balance the number of categories. The number of images in the extended training set is 17360, and the number of images in the test set is 1500, and the extended dataset is named CCTSDB-N.

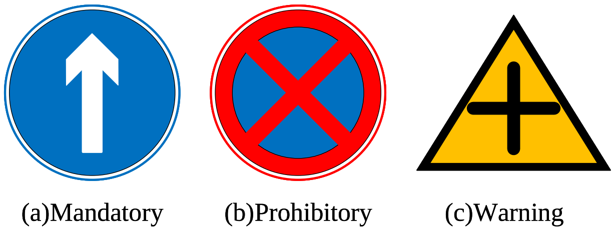


Fig. 9. Classification of the CCTSDB

To verify the generalization of the model on different datasets, TT100K 2021 [41] was chosen as the auxiliary dataset for comparison experiments. The images in TT100K have a resolution of 2048*2048 and contain more small targets with 240 categories of traffic signs. However, some categories have only a small number of instances, so the categories with more than 100 instances are extracted using a Python program.The processed dataset contains 45 categories of traffic signs, and the number of images in the training set is 7222, and the number of images in the test set is 1948.

C. Evaluation Index

The evaluation metrics referenced in the experiments include precision, recall, mean average precision (mAP), average precision (AP), the number of parameters, and the number of floating point operations per second (FLOP). The formulas for precision and recall are shown in (15) and (16), where TP and TN represent the positive and negative samples with correct predictions, and FP and FN represent the positive and negative samples with incorrect predictions, respectively.AP denotes the precision of a single category, and mAP denotes the average precision over all categories, and the computation of AP and mAP is shown in (17) and (18), respectively. FLOPs denotes the computational amount of the model, which is used to measure the computational complexity of the model.

$$P = \frac{TP}{TP + FP} \tag{15}$$

$$R = \frac{TP}{TP + FN} \tag{16}$$

$$AP = \int_0^1 P\left(r\right) dr \tag{17}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{18}$$

D. Comparison of Feature Extraction Modules

To verify the superiority of RepNCSPELAN in feature extraction networks. Comparison experiments are conducted by replacing other modules, the experiments are based on the CCTSDB-N dataset. The experiment compares four modules, namely C2f, OREPANCSPELAN [42], DBBNCSPELAN [43] and DRBNCSPELAN [44], and the experimental results are shown in Table I.

E. Loss Function Comparison

In order to test the effectiveness of the loss function MPDIoU, experiments are conducted on the CCTSDB-N dataset, comparing the five loss functions, CIoU, EIoU, SIoU, WIoU, and MPDIoU. The results of the comparison are shown in Table II.

As evidenced by the experimental results presented in Table II, the MPDIoU loss function outperforms other loss functions with respect to all indices. Compared with the CIoU loss function, the mAP50 improves by 0.6%, the precision improves by 3.4%, and the recall improves by 1.1%. These findings substantiate the efficacy and superiority of the loss function MPDIoU in the domain of traffic sign detection.

F. Compared with Other Advanced Algorithms

To verify the superiority of the improved model, it is compared with other advanced traffic sign detection algorithms. Experiments are conducted on the CCTSDB-N and TT100K datasets, respectively, to verify the robustness of the model under different datasets. The improved model is compared with the other algorithms: YOLOv3, YOLOv7-tiny, YOLOv5s, YOLOv8n, and YOLOv8s. The results of these comparisons are presented in Table III and Table IV.

From Table III, it can be seen tha the improved model exhibits a mAP50 of 84.2%, a mAP50-95 of 54.1%, FLOPs of 10.7G, a model size of 2.8M, and an accuracy and recall of 88.2% and 77.1%, respectively. In comparison with other prevalent algorithms, the improved model exhibits the optimal mAP50, which is 5% superior to YOLOv8n, and mAP50-95 is only surpassed by YOLOv8s. However, the FLOPs are 37.6% of YOLOv8s and the model size is only 12.6% of YOLOv8s. The above comparison indicates that the improved model outperforms other state-of-the-art algorithms.

To verify the performance effect of the improved model on different datasets, experiments are conducted on the TT100K 2021 dataset. From the experimental results in Table IV, it can be seen that although the YOLOv3 model achieves the best accuracy, the computational complexity and model size of the model are much higher than other algorithms, and it is not suitable for traffic sign detection and recognition tasks. YOLOv5s and YOLOv8s have much higher computational complexity and model size than the improved model,

TABLE I
COMPARISON OF FEATURE EXTRACTION MODULES

Models	mAP50(%)	P(%)	R(%)	FLOPs	Params (M)
C2f	81.3	88.6	73.1	12G	3.5
OREPANCSPELAN	81.6	84.8	72.4	10.7G	2.94
DBBNCSPELAN	81.9	88	74.2	10.7G	2.96
DRBNCSPELAN	83.2	87	76.2	10.7G	2.83
RepNCSPELAN	**83.4**	**89.1**	74.4	**10.7G**	**2.8**

TABLE II
LOSS FUNCTION COMPARISON

Models	mAP50(%)	mAP50-95(%)	P(%)	R(%)
CIoU	83.6	53.5	84.8	76
EIoU	80.4	51.1	87.4	72.9
SIoU	81.8	52.2	84.4	72.8
WIoU	80.6	52.8	87	74.3
MPDIoU	**84.2**	**54.1**	**88.2**	**77.1**

TABLE III
COMPARATIVE EXPERIMENTS ON CCTSDB-N

Models	mAP50(%)	mAP50-95(%)	FLOPs	Params (M)	P(%)	R(%)
YOLOv3	82.7	53.3	154.6G	117	89.2	76.7
YOLOv7-tiny	68.8	39.7	13.2G	11.6	81.7	60.1
YOLOv5s	80.7	52.3	15.8G	13.6	89.3	74.2
YOLOv8n	79.2	50.4	8.1G	6	87.4	72.2
YOLOv8s	84.1	54.6	28.4G	21.4	91.7	75.3
Ours	**84.2**	54.1	10.7G	**2.8**	88.2	77.1

TABLE IV
COMPARATIVE EXPERIMENTS ON TT100K

Models	mAP50(%)	mAP50-95(%)	FLOPs	Params (M)	P(%)	R(%)
YOLOv3	91.2	70.8	155.3G	118	88.7	88.1
YOLOv7-tiny	77.8	58.5	13.4G	12	76.9	71.7
YOLOv5s	85.2	65.1	16.1G	14	85.4	78.7
YOLOv8n	79.6	60.8	8.1G	6	78.4	72.1
YOLOv8s	87.1	68.7	28.5G	22	87.6	78
Ours	85.1	66.4	10.9G	3	83.8	77.2

although their accuracy is slightly higher than that of the improved model. Compared with YOLOv8n, the improved model increases the mAP50 by 5.5%, the model size is reduced by half, and the FLOPs increase by only 2.8 G. Compared with the other models, the improved model has the best combined performance in terms of detection accuracy, computational complexity, and model size. The experiments on the TT100K dataset can prove that the improved model has excellent performance under different traffic sign datasets and can be used for traffic sign detection and recognition tasks.

In terms of performance, computational resource requirements and model size, the improved model has clear advantages in the traffic sign detection task, which is suitable for various application scenarios, including resource-constrained mobile and embedded devices, and can provide an efficient and accurate solution for traffic sign detection applications.

G. Ablation Study

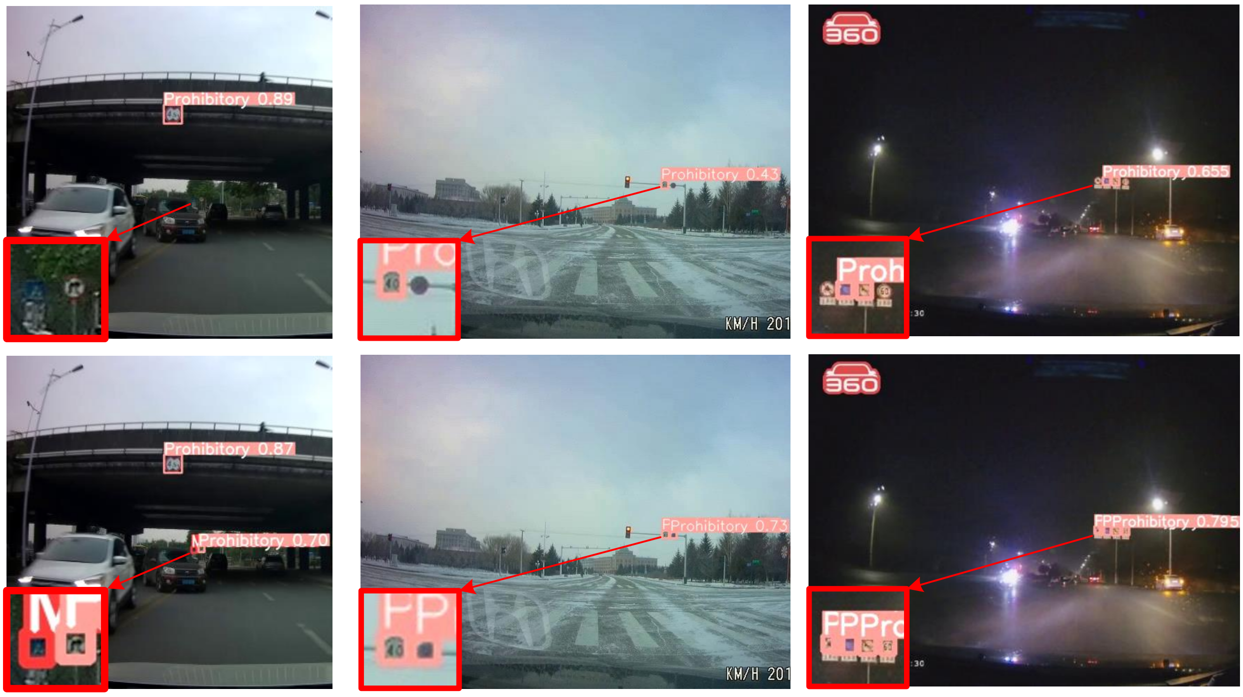
To verify the effectiveness of the improved method, ablation experiments are conducted based on the CCTSDB-N dataset using the YOLOv8n algorithm as a baseline, and the results of the ablation experiments are shown in Table V.

As can be seen from Experiment A, by using the reconstructed feature fusion network, the mAP50 of the model is increased by 3%, and the model size is reduced by 1.5M. This indicates that combining the BiFPN network structure to increase the focus on large-scale features can effectively increase the detection effect of the model on multi-scale targets. Although the computational complexity of the model increases, by adjusting the number of channels, the parameters in the process of feature fusion number is reduced, making the model lighter.

According to Experiment B in Table V, it shows that the mAP50 and mAP50-95 of the model are improved by 1.3% and 1%, respectively, the FLOPs are reduced by 1.5G, and the size of the model is reduced by 0.5M after replacing the C2f module of the feature extraction network with the RepNCSPEALN module. This indicates that the lightweight feature extraction module RepNCSPELAN can effectively improve the feature extraction ability of the model, and reduce the computational complexity and number of param-

TABLE V
ABLATION EXPERIMENTS ON THE CCTSDB-N

Models	P2-BiFPN	RepNCSPELAN	LGSCD	MPDIoU	mAP50(%)	mAP50-95(%)	FLOPs	Params (M)
YOLOv8n					79.2	50.4	8.1G	6
A	✓				82.2	52.4	16.9G	4.5
B	✓	✓			83.5	53.4	15.4G	4
C	✓	✓	✓		83.6	53.5	10.7G	2.8
D	✓	✓	✓	✓	**84.2**	**54.1**	**10.7G**	**2.8**



(a) Daytime (b) Snowy weather (c)Evening

Fig. 10. Comparison of YOLOv8n (top) and RBL-YOLOv8 (bottom) detection results

eters of the model, making the model more lightweight.

From Experiment C, it can be seen that the average accuracy of the model remains unchanged by using the improved lightweight detection head LGSCD, but the FLOPs are reduced by 4.7 G, and the model size is reduced by 1.2 M. This indicates that LGSCD reduces the computational complexity and the parameters of the model by utilizing the strategy of shared convolution, which further lightens the model, and improves model positioning and classification capabilities by using GroupNorm.

As can be seen from Experiment D, by replacing the loss function CIoU with the MPDIoU, the mAP50 and mAP50-95 of the model are both improved by 0.6%, and the computational complexity and the parameters remain unchanged. This indicates that the loss function MPDIoU effectively improves the regression of the bounding box by minimizing the Euclidean distance between the diagonals of the predicted box and the ground truth box, and improves the overall detection of the model.

Compared to the original YOLOv8n algorithm, the ablation experiments show that the improved model improves mAP50 by 5%, mAP50-95 by 3.7%, the size of the model is reduced by half and the FLOPs increase by only 2.6G. The improved model improves the detection accuracy, reduces the complexity of the model, realizes the lightweight of the model, and verifies the effectiveness of the proposed module.

H. Visualization Analysis

To demonstrate the detection effect of the improved model more intuitively, the detection results of the original YOLOv8n algorithm (top panel) and RBL-YOLOv8 (bottom panel) are visualized and compared, revealing the advantages and performance enhancement of the improved model relative to YOLOv8n in different scenarios by comparing the results in three scenarios, namely, daytime, snowy day, and evening, and the detection results are shown in Fig. 10.

As can be seen in Fig. 10, the original YOLOv8n algorithm misses detection in the daytime, snowy, and evening scenarios, and the improved model detects more traffic signs. In the daytime scenario, the original YOLOv8 lost the distant traffic sign targets, and the improved model enhanced the detection of small targets. In the snowy and evening scenarios, the original YOLOv8n algorithm misses the detection of adjacent traffic signs, and the improved algorithm accurately detects and recognizes adjacent traffic signs.

In conclusion, the RBL-YOLOv8 demonstrates superior performance compared to the YOLOv8n algorithm in diverse scenarios. It reduces missed detections, exhibits enhanced detection accuracy, stronger robustness, and superior adaptability, providing a more reliable solution for traffic sign detection.

V. Conclusion

A lightweight multi-scale traffic sign detection and recognition method, RBL-YOLOv8, is presented to reduce the computational complexity and parameters of the model and improve the detection accuracy. Specifically, the RepNCSPELAN module is utilized to improve the feature extraction network, which improves the detection accuracy and reduces the size of the model. Combined with the BiFPN idea, the feature fusion network is reconstructed, which improves the utilization rate of the network on the shallow features and improves the detection of multi-scale targets. The lightweight detection head LGSCD is utilized to reduce the computational complexity and the number of parameters in the detection head part, which achieves lightweight while improving the ability of detection head localization and classification. Finally, the loss function MPDIoU is used to replace CIoU, which enhances the localization ability of the model, accelerates the convergence speed, and improves the detection accuracy. Experiments are conducted on the CCTSDB-N and TT100K datasets, respectively, and the experimental results show that the mAP50 of the improved model reaches 84.2% and 85.1%, respectively, which is 5% and 5.5% higher than the original YOLOv8n algorithm, and the detection effect in different scenarios has been significantly improved. The size of the improved model is 2.8M, which is reduced to half of the original model, and lays a good foundation for subsequent deployment to mobile devices and edge devices. In future work, we will further to improve the detection ability of the RBL-YOLOv8 model for traffic signs and improve it for difficult samples such as occlusion and blur.

References

[1] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-View Traffic Sign Detection, Recognition, and 3D Localisation," *Machine Vision and Applications*, vol. 25, pp. 633–647, 2014.

[2] D. G. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[3] D. Navneet, "Histograms of Oriented Gradients for Human Detection," in *International Conference on Computer Vision & Pattern Recognition, 2005*, vol. 2, 2005, pp. 886–893.

[4] A. De La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road Traffic Sign Detection and Classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[5] J. Miura, T. Kanda, and Y. Shirai, "An Active Vision System for Real-Time Traffic Sign Recognition," in *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.00TH8493)*, 2000, pp. 52–57.

[6] A. Ellahyani, M. El Ansari, R. Lahmyed, and A. Trémeau, "Traffic Sign Recognition Method for Intelligent Vehicles," *JOSA A*, vol. 35, no. 11, pp. 1907–1914, 2018.

[7] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "A Hybrid Vehicle Detection Method based on Viola-Jones and HOG+ SVM from UAV Images," *Sensors*, vol. 16, no. 1325, pp. 1–23, 2016.

[8] A. Møgelmose, D. Liu, and M. M. Trivedi, "Detection of US Traffic Signs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3116–3125, 2015.

[9] L. Deng, O. Abdel-Hamid, and D. Yu, "A Deep Convolutional Neural Network Using Heterogeneous Pooling for Trading Acoustic Invariance with Phonetic Confusion," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6669–6673.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[12] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot Multibox Detector," in *Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.

[16] J. Jeon, B. Jeong, S. Baek, and Y.-S. Jeong, "Static Multi Feature-Based Malware Detection Using Multi SPP-net in Smart IoT Environments," *EEE Transactions on Information Forensics and Security*, vol. 19, pp. 2487–2500, 2024.

[17] J. Wang, Y. Chen, Z. Dong, and M. Gao, "Improved YOLOv5 Network for Real-Time Multi-Scale Traffic Sign Detection," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7853–7865, 2023.

[18] J. Sun and Z. Wang, "Vehicle And Pedestrian Detection Algorithm Based on Improved YOLOv5," *IAENG International Journal of Computer Science*, vol. 50, no. 4, pp. 1401–1409, 2023.

[19] S. Zhou, H. Zhu, X. Liu, Q. Hu, H. Lu, and Z. Peng, "Wood Surface Defect Detection Based on Improved YOLOv8s," *IAENG International Journal of Computer Science*, vol. 51, no. 3, pp. 186–194, 2024.

[20] L. J. Zhang, J. J. Fang, Y. X. Liu, H. Feng Le, Z. Q. Rao, and J. X. Zhao, "CR-YOLOv8: Multiscale Object Detection in Traffic Sign Images," *IEEE Access*, vol. 12, pp. 219–228, 2023.

[21] N. U. A. Tahir, Z. Long, Z. Zhang, M. Asim, and M. ELAffendi, "PVswin-YOLOv8s: UAV-Based Pedestrian and Vehicle Detection for Traffic Management in Smart Cities Using Improved YOLOv8," *Drones*, vol. 8, no. 84, pp. 1–20, 2024.

[22] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic Sign Detection via Improved Sparse R-CNN for Autonomous Vehicles," *Journal of Advanced Transportation*, vol. 2022, pp. 1–16, 2022.

[23] X. Yuan, A. Kuerban, Y. Chen, and W. Lin, "Faster Light Detection Algorithm of Traffic Signs Based on YOLOv5s-a2," *IEEE Access*, vol. 11, pp. 19 395–19 404, 2022.

[24] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *arXiv*, 2024.

[25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and Efficient Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 781–10 790.

[26] Y. Wu and K. He, "Group Normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[27] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A Unified Embedding for Face Recognition and Clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[28] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding YOLO Series in 2021," *ArXiv Preprint ArXiv:2107.08430*, 2021.

[29] 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 Transactions on Cybernetics*, vol. 52, no. 8, pp. 8574–8586, 2021.

[30] M. Siliang and X. Yong, "Mpdiou: A Loss for Efficient and Accurate Bounding Box Regression," *ArXiv*, 2023.

[31] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of YOLO Algorithm Developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[32] 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," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[33] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[34] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[35] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 002–21 012, 2020.

[36] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, “Tood: Task-Aligned One-Stage Object Detection,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 3490–3499.

[37] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A New Backbone That Can Enhance Learning Capability of CNN,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 390–391.

[38] C. Wang, H. Liao, and I. Yeh, “Designing Network Design Strategies Through Gradient Path Analysis,” *Journal of Information Science and Engineering*, vol. 39, no. 2, pp. 975–995, 2023.

[39] A.-F. O. Detector, “FCOS: A Simple and Strong Anchor-Free Object Detector,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1922–1933, 2022.

[40] J. Zhang, X. Zou, L.-D. Kuang, J. Wang, R. S. Sherratt, and X. Yu, “CCTSDB 2021: A More Comprehensive Traffic Sign Detection Benchmark,” *Human-Centric Computing and Information Sciences*, vol. 12, pp. 1–18, 2022.

[41] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-Sign Detection and Classification in the Wild,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2110–2118.

[42] M. Hu, J. Feng, J. Hua, B. Lai, J. Huang, X. Gong, and X.-S. Hua, “Online convolutional re-parameterization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 568–577.

[43] X. Ding, X. Zhang, J. Han, and G. Ding, “Diverse Branch Block: Building a Convolution as an Inception-Like Unit,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 886–10 895.

[44] X. Ding, Y. Zhang, Y. Ge, S. Zhao, L. Song, X. Yue, and Y. Shan, “Unireplknet: A Universal Perception Large-Kernel Convnet for Audio, Video, Point Cloud, Time-Series and Image Recognition,” *ArXiv*, 2023.