IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 September 2023, accepted 6 October 2023, date of publication 11 October 2023, date of current version 19 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3323618

APPLIED RESEARCH

Sign-YOLO: A Novel Lightweight Detection Model for Chinese Traffic Sign

WEIZHEN SONG AND SHAHREL AZMIN SUANDI, (Senior Member, IEEE)
Intelligent Biometric Group, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Engineering Campus, Nibong Tebal, Penang 14300, Malaysia

Corresponding author: Shahrel Azmin Suandi (shahrel@usm.my)

ABSTRACT Traffic sign recognition plays a crucial role in the intelligent vehicle's environment perception system. However, due to varying weather conditions, illumination, and complicated backgrounds, recognizing traffic signs becomes very challenging. A novel lightweight detection model based on YOLOv5s, namely Sign-YOLO, is proposed to overcome these challenges. Firstly, the CA (Coordinate Attention) module is incorporated into the backbone network to improve the extraction of key features. Secondly, the improved High-BiFPN is used to enhance YOLOv5s' neck structure's capability in fusing multi-scale semantic information. Finally, the improved Better-Ghost Module is employed to reduce the model's parameters and accelerate the detection speed. We used the CCTSDB2021 dataset to evaluate our model. Compared to YOLOv5s, the proposed Sign-YOLO algorithm in this paper reduces the model parameters by 0.13 M. The precision, recall, F-1 score, and mAP value have improved by 1.02%, 7.01%, 1.84%, and 4.61%, respectively. The FPS value remains around 86 fps. The results show that Sign-YOLO has achieved the optimal balance between accuracy and real-time performance.

INDEX TERMS Chinese traffic sign, intelligent vehicle, deep learning, lightweight model, YOLOv5s.

I. INTRODUCTION

Traffic sign recognition is a crucial research area in intelligent transportation. Its goal is to provide drivers with valuable traffic information to enhance their driving safety [1]. However, different weather conditions, such as clouds, snow, and fog, as well as varying illumination during the day and night, along with complicated backgrounds, often increase the difficulty of traffic sign recognition in vehicle driving [2]. Therefore, locating and identifying traffic signs in real-world scenarios remains a challenging task.

Computer vision-based traffic sign detection algorithms can be mainly divided into two categories: traditional detection algorithms and deep learning-based detection algorithms. Traditional traffic sign detection algorithms typically involve a sequential set of steps to detect traffic signs. Firstly, they segment the regions of interest containing traffic signs. Then, they employ manual feature extraction methods to extract information from these regions based on color, shape, and pixel values using techniques such as LBP [3], SIFT [4], and HOG [5]. Finally, machine learning approaches like Random Forest [6], Adaboost [7], and SVM [8] are utilized for the classification of the detected traffic signs. However, manual feature extraction methods face challenges in capturing complex semantic information from images, which leads to limited expressive power in representing image features and poor robustness across various tasks.

The popularity of deep learning-based detection models has gradually increased due to the constraints imposed by traditional methods. CNNs are employed to train these models, utilizing a vast number of images in the process. The continuous evolution of network parameters enables the models to effectively capture traffic sign characteristics. There are two main types of models used in deep learning: two-stage models and one-stage models. In the case of two-stage models, the first step involves a search to detect traffic signs and identify regions of interest. Once these regions are located, a feature extraction network is employed to obtain the coordinates and categories of the traffic signs. Representative examples of this category include the R-CNN series [9], [10], [11] and Mask

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

R-CNN [12], etc. In one-stage models, they predict both the class probabilities and positional coordinates of traffic signs at the same time. The SSD series [13], [14] and the YOLO series [15], [16], [17], [18], [19], [20] are representative models of this category. These detection models achieve high accuracy but suffer from large parameters and slow detection speeds.

Selecting the lightweight detection model YOLOv5s is a recommended strategy for resolving issues related to large parameters and slow detection speeds in traffic sign detection models. Although it has fewer parameters and faster detection speed, its detection accuracy is relatively lower. Therefore, this paper focuses on enhancing the network structure of YOLOv5s, and the main contributions of Sign-YOLO compared to the original version are listed as follows.

- By incorporating a coordinate attention mechanism into the backbone network, the network becomes more focused on interesting regions, thus enhancing its ability to extract crucial features and making the model more proficient at detecting traffic signs in real-world scenarios.
- The proposed High-BiFPN enhances PANet's capability to fuse features from multi-scale objects by employing a weighted bi-directional feature pyramid network with intra-cell skip connections.
- The proposed HAG(High Active Ghost) module aims to reduce redundancy in feature maps using cheap operations, thereby reducing the parameters of YOLOv5s.

II. RELATED WORK

Using Google Scholar, we searched for recent publications on traffic sign detection and recognition. After extensive research, we have classified the techniques into two categories: traditional approaches and deep learning-based methods.

Dai et al. [21] developed a novel strategy to improve the accuracy of traffic sign identification under varying lighting conditions. They achieved this by leveraging color features, resulting in an impressive accuracy of 78% and a processing speed of 11 fps. Liang et al. [22] introduced a pioneering method for traffic sign recognition, which integrated HOG-Gabor feature extraction and a fusion of Softmax classifiers. This novel approach demonstrated a remarkable accuracy of 97.68%. Xu et al. [23] proposed a novel traffic sign detection method that combines adaptive color thresholding segmentation and shape symmetry hypothesis testing. This approach effectively utilizes both the available traffic sign information and the image data to enhance the accuracy of the detection process. The initial stage required the computation of an adaptive segmentation threshold through an examination of the cumulative distribution function derived from the image histogram. The resulting thresholded image exhibited a distinct shape characteristic, which was further transformed into a feature vector representing interconnected regions. This method achieved a traffic sign detection accuracy surpassing 94%. Sun et al. [24] introduced a traffic sign detection approach that utilizes adaptive gamma correction. Their method yielded impressive results, with a detection rate of 97.28% and a false detection rate of 10.35%. Calero et al. [25] used HOG features and an ELM classifier to detect and recognize traffic signs under extreme daytime lighting conditions, achieving an accuracy of 96.71% and a detection speed of 24 fps. Wang et al. [26] proposed a fast and accurate localization of moving targets based on FrFT. Traditional approaches require manual extraction of color characteristics, shape characteristics, or a combination of both to extract precise information from traffic signs. However, these methods are susceptible to interference from environmental factors such as variations in lighting conditions, severe weather, and complex backgrounds.

The need for manual feature extraction is eliminated with the utilization of deep learning-based techniques in traffic sign detection. Deep learning models are trained using a substantial amount of labeled data samples, enabling them to acquire knowledge of nonlinear functions. These functions convert images into a feature space where linear classifiers can quickly distinguish between classes, resulting in accurate traffic sign recognition. Cui et al. [27] introduced the CAB Net, an innovative method aimed at improving the accuracy of traffic sign detection. The primary emphasis of this method was on generating high-resolution and reliable semantic feature maps. The outcomes were exceptionally impressive, achieving an mAP of 89% while simultaneously maintaining a swift detection speed of 27 fps. Li et al. [28] combined MobileNet with Faster R-CNN to improve the detection accuracy of small traffic signs by integrating color and shape attributes. For the classification of traffic signs, they adopted an efficient CNN using asymmetric kernels. The experimental outcomes demonstrated the effectiveness of their proposed detector, successfully identifying traffic signs across all categories. Dewi et al. [29] employed GANs to produce a larger and more diverse collection of training images, thereby enhancing the dataset. By integrating synthetic images with the original ones, they aimed to enhance the overall quality of the dataset. The outcomes of their study indicated an accuracy of 84.9% using YOLOv3 and a further improvement to 89.33% with YOLOv4. Ayachi et al. [30] presented a traffic sign detection model employing the YOLO approach. This model exhibited impressive results by employing both model quantization and pruning techniques, achieving a remarkable mAP score of 96% while maintaining a fast-processing speed of up to 16 fps. Lu et al. [31] proposed STDN, a traffic sign detection network that combines PosNeg balanced anchors and domain adaptation techniques. This network achieves a detection speed of 55.9 fps. Liang et al. [32], [33] proposed an improved Sparse R-CNN, which combines coordinate attention blocks with ResNeSt to construct a feature pyramid and modify the backbone network. This modification allows the extracted features to focus on crucial information, thereby enhancing the accuracy of traffic sign detection. Subsequently, they introduced the DetectFormer algorithm,

achieving detection performance of 97.6% AP50 and 91.4% AP75 on the BCTSDB dataset.

Deep learning-based models have demonstrated remarkable effectiveness in improving the performance of traffic sign detection. However, finding a balance between enhancing the precision of traffic sign detection and maintaining a compact model size remains a significant challenge. Therefore, we have designed and implemented Sign-YOLO, a model for traffic sign detection that offers high precision, a lightweight design, and strong robustness.

III. METHOD

We will discuss YOLOv5s and the improved Sign-YOLO algorithm. Firstly, we have incorporated the coordinate attention module into the backbone network of YOLOv5s. Next, we have enhanced the structure of the neck network in YOLOv5s by integrating an improved High-BiFPN. Lastly, we have reduced the number of parameters in the Sign-YOLO model by utilizing the improved Better-Ghost module.

A. YOLOv5s NETWORK MODEL

YOLOv5s is designed as a one-stage object detection model that follows a specific methodology. The central concept involves the extraction of image features through a backbone network. These features are subsequently processed by the neck network structure. Finally, in the output layer, the model performs classification and regression tasks to predict the bounding boxes of objects along with their associated class confidences. The network structure of YOLOv5s is shown in Figure 1.

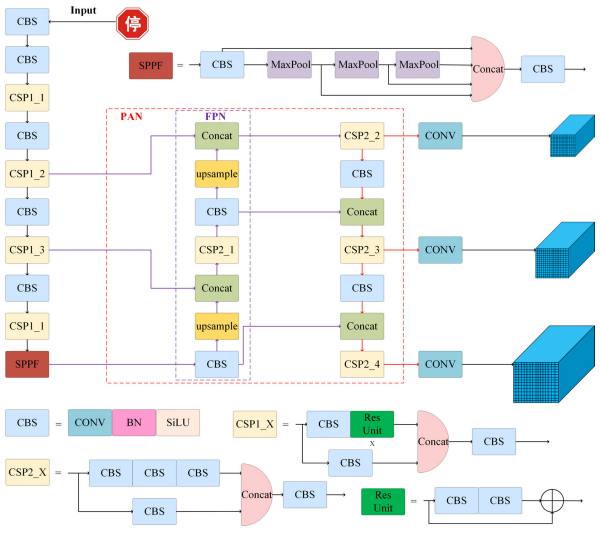

FIGURE 1. YOLOv5s network structure.

The YOLOv5s network architecture consists of three main components: the backbone, neck, and head. (1) Feature information is extracted from images mainly through the backbone network, which includes three critical modules: the CBS module, the CSP module, and the SPPF module. (2) Figure 2 illustrates the neck architecture, comprising two essential

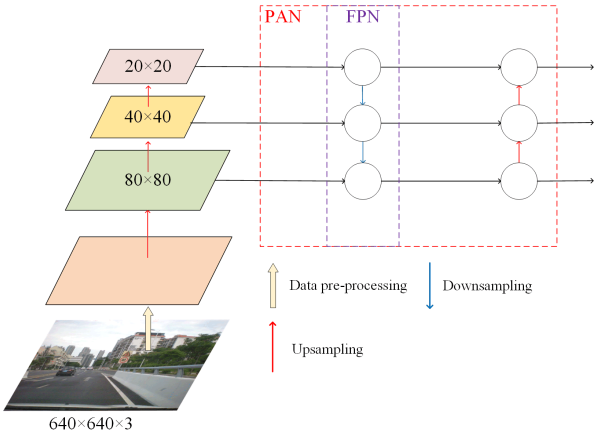

FIGURE 2. PAN network structure.

elements: the FPN structure and the PAN structure. The PAN structure consists of an FPN structure and a bottom-up pyramid structure, which can fuse top-down semantic information and bottom-up positional information. These pieces of information are derived from feature maps extracted by the backbone network at multiple scales. (3) The head section utilizes the fused feature results from the neck module to make predictions, resulting in three different prediction scales: 20×20, 40×40, and 80×80.

B. CONSTRUCTION OF THE SIGN-YOLO

After introducing the principle of how YOLOv5s detects objects, we will describe the Sign-YOLO algorithm that we have proposed in this section.

1) STEP 1:COORDINATE ATTENTION MECHANISM

Integrating an attention mechanism module into YOLOv5's backbone network gives the model the ability to focus on relevant details of interesting traffic signs. This minimizes the influence of the surrounding environment and improves the model's detection accuracy.

The representative attention mechanisms include the BAM [34], CBAM [35], SE [36], and coordinate attention module (CA) [37]. While SE attention primarily focuses on capturing inter-channel information, it may overlook the significance of positional information. The CA module, on the other hand, serves as a high-performing and lightweight attention mechanism. To begin, the channel attention mechanism is restructured by dividing it into two parallel 1D feature encodings. Two separate 1D global pooling operations are employed to encode input features independently, with one operating vertically and the other horizontally. This approach generates two separate feature maps that are sensitive to their respective directions. In this way, it embeds spatial coordinate position information into the channel attention mechanism, thereby obtaining precise location data while capturing long-range dependencies. Subsequently, the two feature maps with unique directional information undergo a conversion process, resulting in the generation of two attention maps. These attention maps are then used to adjust the input feature map

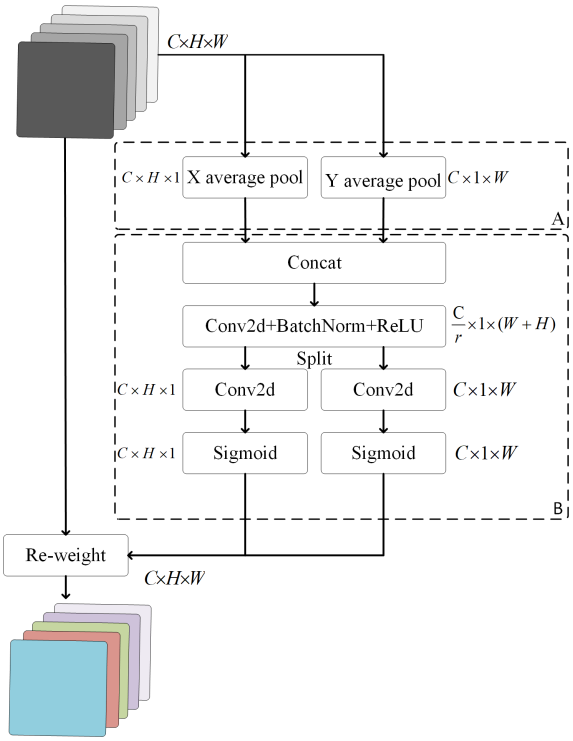

FIGURE 3. The network structure of CA module.

by assigning varying weights to its elements based on their importance for position and direction. This has the potential to improve the model's capability in accurately recognizing and detecting targeted traffic signs. The network structure of the CA module is shown in Figure 3.

Global pooling is split into two 1D feature encodings, where an input X undergoes pooling operations with kernels of sizes $(H, 1)$ and $(1, W)$ along the horizontal and vertical dimensions, respectively. These pooling operations encode each channel individually. For a given position (a, b) and channel c, the pixel value of channel c is represented as $x_c(a, b)$. As a result, the average pooling value in the horizontal direction can be formulated as follows:

$$p_c^a(a) = \frac{1}{W}\sum_{0\le b\le W} x_c(a, b). \tag{1}$$

Similarly, the pooling values in the vertical direction can be written as:

$$p_c^b(b) = \frac{1}{H}\sum_{0\le a\le H} x_c(a, b). \tag{2}$$

Afterward, the obtained feature maps are combined by concatenating them and applying a shared 1×1 convolutional operation for further transformation. The relevant definition is as follows:

$$z = \theta(q([p_c^a, p_c^b])). \tag{3}$$

where $[,]$ represents the concatenation operation, q represents the convolution operation, θ represents the SiLU activation function, and $z \in \mathbb{R}^{(\frac{C}{r})\times(H+W)}$ represents the intermediate feature map encoding spatial information in both horizontal and vertical directions.

Then, the tensor z can be split into two separate tensors, denoted as z^a and z^b, by performing a division along the spatial dimension. Afterward, the feature maps z^a and z^b are subjected to transformations using two 1×1 convolutions, q_a and q_b, respectively. These convolutions result in tensors with the same number of channels as the input X.

$$u^a = \sigma(q_a(z^a)). \tag{4}$$

$$u^b = \sigma(q_b(z^b)). \tag{5}$$

where σ represents the sigmoid activation function. Therefore, the feature map output of the CA module is defined as

$$f_c(a, b) = x_c(a, b) \times u_c^a(a) \times u_b^b(b). \tag{6}$$

where c represents the number of channels. $u_c^a(a)$ represents the weight at the a-th position in the W direction, while $u_b^b(b)$ represents the weight at the b-th position in the H direction. $x_c(a, b)$ denotes the value of the input feature map, and $f_c(a, b)$ represents the value of the output feature map. Figure 4 illustrates the incorporation of the CA module into the CSP structure of the YOLOv5s backbone network.

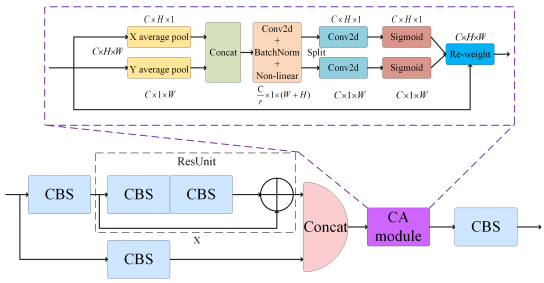

FIGURE 4. The structure of CA_CSP module.

2) STEP 2:MULTI-SCALE FEATURE LAYER FUSION MODULE HIGH-BIFPN

YOLOv5s employs the PAN structure to fuse input features at multiple scales. The input features are simply added together without distinction in this way. However, the contributions of the fused output features are often unequal due to the differences in resolutions among the input features. To address this problem, Tan et al. [38] proposed a Weighted Bi-directional Feature Pyramid Network (BiFPN). Weighted fusion is a mechanism employed by BiFPN to understand the significance of various input features. According to this strategy, each node in the network assigns weights to the input features and trains these weights using a fast normalization method, as shown in Equation 7.

$$\text{Out} = \sum_i \frac{w_i I_i}{\varepsilon + \sum_j w_j}. \tag{7}$$

where I_i represents the input feature, Out represents the result of weighted feature fusion, w_i and w_j are learnable weights. The ReLU activation function is utilized to constrain the learnable weights to the range of $[0, 1]$. $\varepsilon = 0.0001$ is a small value to ensure output stability.

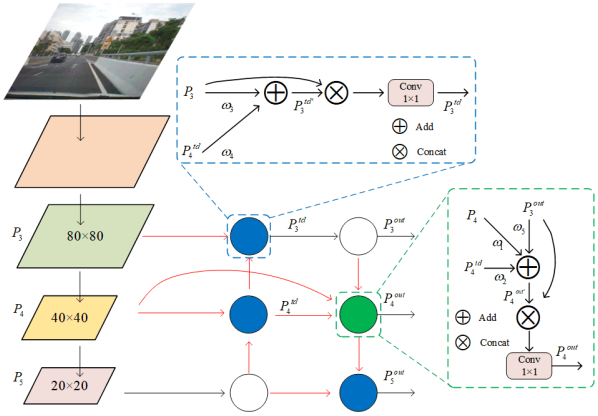

FIGURE 5. High-BiFPN network structure.

This paper is inspired by the ideas of BiFPN. In the YOLOv5s model, after a 16-fold downsampling, the resulting 40×40 feature map is connected to the subsequent feature maps through cross-layer connections. This enhancement improves the network's ability to detect traffic signs by allowing for a more thorough extraction of positional information, minimizing the loss of feature information, and enhancing overall performance. The improved High-BiFPN structure is shown in Figure 5.

The blue nodes represent nodes with two branching inputs, while the green nodes represent nodes with three branching inputs. Weights are assigned to different input features based on their varying contributions to the output features at different scales. Taking the blue nodes of the P_3 layer as an example, the definition is as follows:

$$P_3^{\text{td}'} = \text{Conv}\left(\frac{w_3 \times P_3 + w_4 \times \text{Resize}(P_4^{\text{td}})}{w_3 + w_4 + \varepsilon}\right). \quad (8)$$

where *Conv* is the convolution operation, *Resize* upsamples the input, w_3 is the learnable weight of the output of the third layer P_3, and w_4 is the learnable weight of the output of the fourth layer P_4^{td}.

In Figure 5, we demonstrate the concatenation of $\left(P_3^{\text{td}'}\right)$ with P_3. Subsequently, a 1×1 convolution is employed to merge the current features into a new feature, resulting in the output denoted as P_3^{td} at this node. This intra-cell jump connection structure serves a dual purpose: it reconstructs the features while also increasing the number of channels, maintaining the original channel count in the YOLOv5s model. This further enhances the feature fusion capability of the feature pyramid. The connection of the intra-cell jump connection structure is illustrated in Equation 9.

$$P_3^{td} = \text{Conv}(\text{concat}(P_3, P_3^{td'})). \quad (9)$$

where *concat* is the concatenation operation and *Conv* is the convolution operation.

When different scale features are inputted into the green node from three branches, we can similarly obtain $P_4^{\text{out}'}$ and output P_4^{out}, as shown in equations 10, 11.

$$P_4^{\text{out}'} = \text{Conv}\left(\frac{w_1 \times P_4 + w_2 \times P_4^{\text{td}} + w_5 \times \text{Resize}(P_3^{\text{out}})}{w_1 + w_2 + w_5 + \varepsilon}\right). \quad (10)$$

$$P_4^{\text{out}} = \text{Conv}(\text{concat}(P_4^{\text{out}'}, \text{Resize}(P_3^{\text{out}}))). \quad (11)$$

where *Resize* is the downsampling operation and w_1, w_2, and w_5 are the learnable weights corresponding to the three input feature maps of $P_4^{\text{out}'}$. We propose High-BiFPN, which improves the concatenation module in the PAN structure of YOLOv5s.

3) STEP 3: THE LIGHTWEIGHT BETTER-GHOST MODULE

In traditional convolutional methods, all channels of the input feature map are simultaneously considered, resulting in the generation of a large number of redundant features in the intermediate feature map. These redundant features help us fully grasp the input data, but they also demand a lot of computational resources. Han et al. [39] proposed a novel lightweight Ghost module that can generate redundant feature maps using cheap operations. This paper, inspired by the Ghost Module, proposes a Better-Ghost Module (BGM). The structure of BGM is depicted in Figure 6. The first part of the BGM is generated by a CBS operation with a size of 1×1 , output channels equal to half of the input channels, and a grouping of 1, producing the real feature layer. Then, the result of the first part of the CBS operation is used as input, and the Ghost feature layer is obtained through a 3×3 CBS operation with the same number of input and output channels and the same number of groups as the input channels. The final step involves combining the output feature maps from the two components and using them as the ultimate output feature map for the BGM. The CBS module consists of a $k \times k$ dimension convolutional layer, a Batch Normalization layer, and a SiLU activation function.

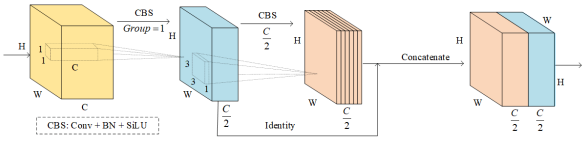

FIGURE 6. The structure of lightweight Better-Ghost Module.

The original Ghost Module used the ReLU activation function, while the BGM structure in this paper uses the SiLU activation function. This choice is made because the ReLU activation function has a gradient of 0 for the negative half of the x-axis. Consequently, negative gradients are set to 0 after passing through ReLU, causing sparsity in negative values. This can lead to the phenomenon of "dead neurons," where neurons become ineffective in learning meaningful features. The SiLU function, on the other hand, maintains a non-zero gradient on the negative half-axis, allowing weights to continue updating on the negative side. This effectively prevents

the loss of negative gradient information and provides higher activation, enabling the extraction of more effective features. As a result, it improves the training efficiency of the model. The two activation function curves are shown in Figure 7.

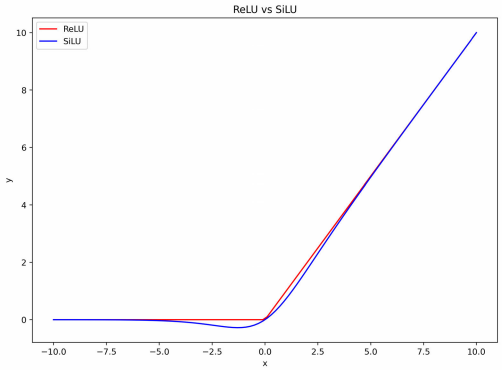

FIGURE 7. ReLU and SiLU activation functions.

We will present evidence from memory usage and computational demands to demonstrate the effectiveness of the BGM module. Assuming the size of the input feature map and output feature map is $C \times H \times W$, where C, H, and W represent the number of channels, height, and width of the feature map, respectively. In the BGM module, the convolutional kernel is $k \times k \times \frac{C}{g}$, where k and g denote the size and number of groups of the convolutional kernel, respectively. The convolution computation is divided into two steps. In the first step, with $k = 1$ and $g = 1$, the real feature map is generated. In the second step, with $k = 3$ and $g = C/2$, the ghost feature map is generated. The flops can be calculated as $\frac{C}{2} \times H \times W \times C \times 1 \times 1 + \frac{C}{2} \times H \times W \times 3 \times 3$, and the parameters can be calculated as $1 \times 1 \times C \times \frac{C}{2} + 3 \times 3 \times 1 \times 1 \times \frac{C}{2}$. In the case of a normal convolution where the input feature map and output feature map are of the same size as the BGM module, the convolutional kernel size is 3×3, the FLOPs are $C \times H \times W \times C \times 3 \times 3$, and the parameters are $C \times C \times 3 \times 3$. The ratio of computational complexity between the BGM module and regular convolution is

$$\gamma = \frac{C \times H \times W \times C \times 3 \times 3}{\frac{C}{2} \times H \times W \times C \times 1 \times 1 + \frac{C}{2} \times H \times W \times 3 \times 3} = \frac{18C}{C+9}. \tag{12}$$

The comparison of parameters between the BGM module and regular convolution is

$$\varphi = \frac{C \times C \times 3 \times 3}{1 \times 1 \times C \times \frac{C}{2} + 3 \times 3 \times 1 \times 1 \times \frac{C}{2}} = \frac{18C}{C+9}. \tag{13}$$

By examining equations (12) and (13), it becomes evident that as the number of channels grows, the ratio experiences a corresponding increase. Therefore, when inputting feature maps of the same size, the BGM module used in this paper requires fewer parameters, resulting in faster processing speed. Moreover, as the number of channels increases, the benefits obtained from the BGM module become more significant. The paper uses an improved BGM module to replace the conventional convolution in the CSP2_4 module of the YOLOv5s network. Figure 8 illustrates the enhanced architecture of the Sign-YOLO model after incorporating the aforementioned improvements.

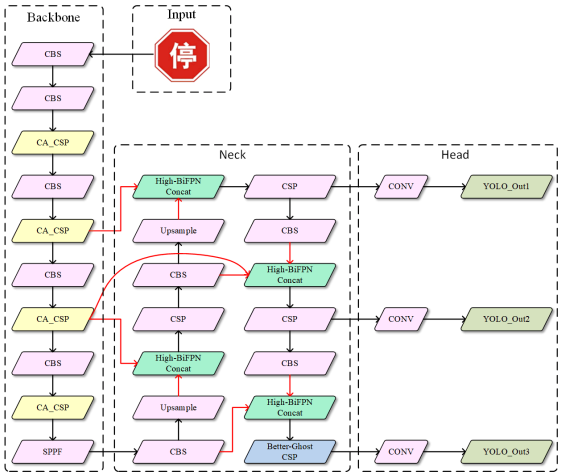

FIGURE 8. Sign-YOLO network structure.

IV. EXPERIMENT

A. DATASET

We will validate the Sign-YOLO model using the CCTSDB2021 dataset. The dataset divides traffic signs into three types: prohibition signs, warning signs, and mandatory signs. It consists of 16,356 images created through manual processing of 423 different videos, captured at various times, weather conditions, locations, and speeds, ensuring data diversity. The images in the dataset have resolutions of 860 × 480, 1280 × 720, 1920 × 1080, etc. The training dataset includes a total of 16,356 images, and for evaluation purposes, it has a separate test set containing 1,500 images. During the training phase, the dataset is divided into two distinct sets: the training set and the validation set, with a 9:1 ratio. Figure 9 displays some sample images from the CCTSDB 2021 dataset.

B. EXPERIMENTAL CONFIGURATION AND PARAMETER SETTINGS

The experiment was conducted on a computer running the Windows 11 operating system. The computer was equipped with an Intel Core i9-13900KF CPU clocked at 3.00 GHz, an NVIDIA GeForce RTX 4070 graphics card, and 12GB of graphics memory. The experiment utilized the Python programming language along with PyTorch 1.13.1 for implementation. To accelerate the training process, CUDA 11.1 was employed. We used the SGD optimizer to update the model parameters during training for 400 epochs, with a batch size of 16. The initial learning rate was set to 0.001, with a applied momentum of 0.937 and weight decay of 0.0005. Additionally, we incorporated mosaic and mixup techniques

FIGURE 9. Some samples in CCTSDB2021.

with a ratio of 80%. For all other configurations, we adhered to the default settings as used in the original YOLOv5s model.

To obtain 9 anchor boxes of various scales, including large, medium, and small, the K-means method was initially applied to analyze the CCTSDB2021 dataset during the training process. The anchor box sizes are as follows: (22, 38), (26, 62), (48, 92), (15, 26), (12, 33), (16, 46), (5, 10), (6, 13), and (10, 19). Then, the model was trained without pretraining using these parameters.

C. EVALUATION METRICS

This study evaluates the algorithm's performance using multiple assessment metrics, including average precision (AP), precision (P), F-1 score, mean Average Precision (mAP), recall (R), and Frames Per Second (FPS). FPS quantifies the rate at which images are processed per second. True Positives (TP) indicate correctly detected traffic signs that match their true meaning. False Positives (FP) refer to cases where a traffic sign is detected, but its meaning does not match the true meaning. False Negatives (FN) represent traffic signs that the model fails to detect. The following formulas can be used to calculate these metrics:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{14}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{15}$$

$$\text{AP} = \int_0^1 P(R)\, d(R). \tag{16}$$

$$\text{mAP} = \frac{1}{\text{classes}} \sum_{i=1}^{\text{classes}} \int_0^1 P(R)\, d(R). \tag{17}$$

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{18}$$

D. RESULTS AND ANALYSIS

1) EVALUATION RESULTS

In this study, we utilize enhanced modules in combination with YOLOv5s to create Sign-YOLO, which is subsequently evaluated and compared with the YOLOv5s algorithm using the CCTSDB2021 dataset. The results achieved are presented in Table 1.

With an IOU threshold set to 0.5, the Sign-YOLO model increased the P-value by 1.02%, the R-value by 7.01%, the F1-score by 4.84%, and the mAP value by 4.61%. Furthermore, the model size decreased by 0.13 M. In this paper, we successfully achieved improvement in various metrics in the Sign-YOLO model, and its parameters are smaller than those of YOLOv5s. The significant improvement in the R-value indicates that Sign-YOLO has a better ability to recognize and locate traffic signs, reducing the occurrence of false negatives (where real targets are incorrectly classified as negative instances). This means that more traffic signs will be accurately detected, contributing to improved traffic safety. To further facilitate a detailed comparison, we have provided the P curves, R curves, F1-score curves, and PR curves for the three traffic sign categories of the CCTSDB2021 dataset, as shown in figure 10.

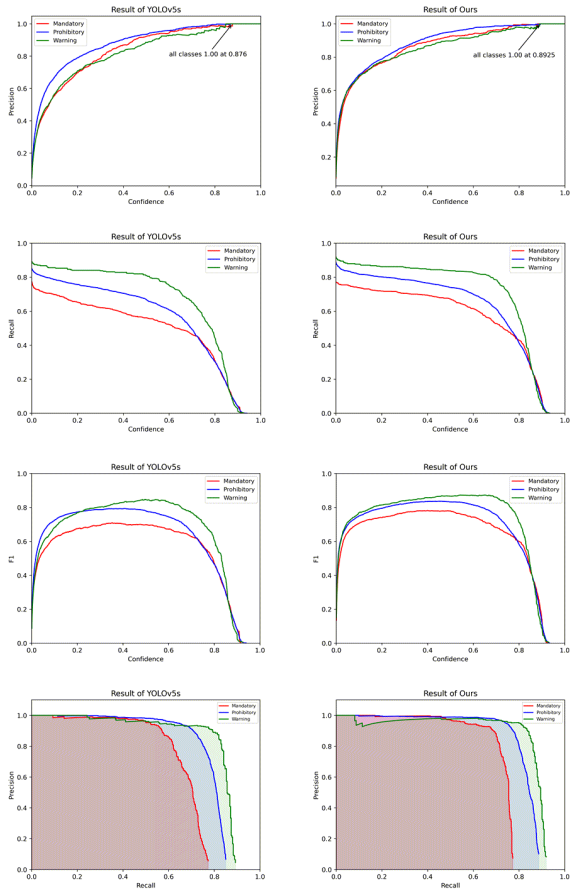

FIGURE 10. Comparison of P curves, R curves, F1 curves, and PR curves for YOLOv5s and Sign-YOLO.

At a confidence score of 0.5, the corresponding P, R, F1, and AP values for the YOLOv5s and Sign-YOLO algorithms are presented in table 2. The Sign-YOLO model has demonstrated remarkable advancements in performance metrics for detecting three classes of traffic signs, surpassing the capabilities of YOLOv5s.

TABLE 1. Comparison results of Sign-YOLO and YOLOv5s on the CCTSDB2021 dataset.

Method	Kmeans	CA module	High-BiFPN	Better-Ghost	P(%)	R(%)	F1(%)	mAP(%)	Size(M)
YOLOv5s					91.48	67.96	77.98	76.39	7.03
a	✓				91.46	73.12	81.26	78.38	7.03
b	✓	✓			92.12	73.45	81.73	80.12	7.10
c	✓	✓	✓		92.27	75.00	82.74	81.33	7.23
Ours	✓	✓	✓	✓	92.50	74.97	82.82	81.00	6.90

TABLE 2. Performance of YOLOv5s and Sign-YOLO on the CCTSDB2021 dataset.

Model	Class	Evaluation Indicator			
		P (%)	R (%)	F1 Score	AP (%)
YOLOv5s	Mandatory	92.43	56.13	0.70	67.63
	Prohibitory	93.52	66.97	0.78	78.53
	Warning	88.49	80.78	0.84	83.01
Ours	Mandatory	92.69	67.13	0.78	73.14
	Prohibitory	95.10	74.00	0.83	83.15
	Warning	89.71	83.78	0.87	86.69

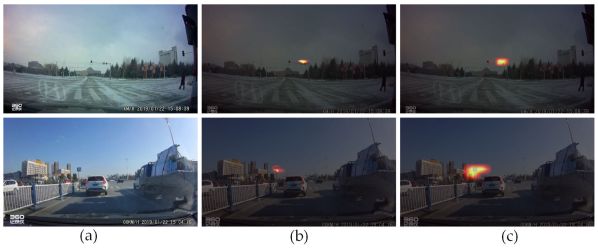

FIGURE 11. Performance comparison of YOLOv5s and ours. (a) original image; (b) results of YOLOv5s; (c) Results of Sign-YOLO.

In this paper, we introduced relevant tests in real-life scenarios. Firstly, we utilized a vehicle-mounted camera to capture video footage of the road environment. The video was processed into multiple images, which were then subjected to testing using the YOLOv5s algorithm and the Sign-YOLO algorithm. The results of these tests were visualized using a heatmap, as shown in Figure 11. The findings demonstrate that the Sign-YOLO model places more emphasis on detecting traffic signs within images and achieves superior performance in terms of recognition. According to these findings, the Sign-YOLO algorithm outperforms the YOLOv5s model in terms of detection and recognition capabilities.

2) PERFORMANCE COMPARISON

Sign-YOLO was thoroughly validated on the CCTSDB2021 dataset through a series of extensive comparative experiments. The paper utilized diverse evaluation metrics to conduct a comprehensive quantitative assessment from multiple viewpoints. The comparative results are presented in table 3.

Sign-YOLO is compared with several object detection algorithms, encompassing both one-stage and two-stage approaches. The evaluation involves well-known techniques such as Faster R-CNN, Libra R-CNN, Dynamic R-CNN, Sparse R-CNN, SSD, RetinaNet, and the YOLO series. Generally, one-stage object detection algorithms demonstrate faster processing times in contrast to their two-stage counterparts, but two-stage algorithms tend to provide higher accuracy. Faster R-CNN is a classic two-stage detection algorithm with P, R, F1 score, and mAP of 84.43%, 54.98%, 0.60, and 56.58% respectively. Recent advancements in two-stage detection include Libra R-CNN, Dynamic R-CNN, and Sparse R-CNN. Sparse R-CNN achieved significant improvements with precision at 94.12%, an F1 score of 0.67, and an mAP of 59.65%, surpassing Faster R-CNN by 9.69%, 7.00%, and 3.07% respectively. However, despite these gains, the detection speed is limited by the two-stage model, and it is challenging to improve the detection speed of this algorithm. As a result, it cannot achieve real-time object detection in various road environments in the wild. SSD has several limitations and cannot accurately detect traffic signs. The RetinaNet algorithm has improved the average detection precision of traffic signs, but the FPS value is 8.88. The YOLO series has made significant advancements since then. YOLOv5s, compared to previous algorithms, has achieved impressive detection results with a mAP value of 76.39%. The model size is 7.03M, and its FPS is 112. Compared to YOLOv5s, the Sign-YOLO algorithm achieves optimization by combining multiple modules, resulting in a higher recall rate, precision, F1 score, and mAP in traffic sign object detection tasks. Additionally, it achieves an FPS value of 86. Therefore, Sign-YOLO effectively meets the detection needs of traffic signs for self-driving vehicles in a variety of road situations by striking an appropriate balance between detection accuracy, speed, and model size.

Detection algorithms exhibit varying performance depending on the constantly changing and complex weather conditions encountered during everyday driving in the wilderness. Table 4 presents the detection outcomes of various algorithms on the CCTSDB2021 test set across five weather conditions, using an IOU threshold of 0.5.

It can be observed that different traffic sign detection algorithms achieve relatively high precision and recall rates in cloudy, snowy, and sunny conditions. This indicates that these algorithms perform well in situations with good lighting and without fog interference. However, in foggy weather and at night, different detection algorithms exhibit relatively low precision and recall rates, suggesting that foggy weather adversely affects the detection of traffic signs. Nighttime conditions have a more significant impact on the detection of traffic signs compared to foggy weather, as visibility is relatively low at night. This is due to the relatively dim

TABLE 3. Detection results of different networks on the CCTSDB2021 dataset.

Model	Evaluation Indicator					
	P (%)	R (%)	F1 Score	mAP@0.5/%	FPS	Size(M)
Faster R-CNN [11]	84.43	54.98	0.60	56.58	4.87	143.70
Libra R-CNN [40]	83.72	60.04	0.70	61.35	8.81	-
Dynamic R-CNN [41]	86.98	58.33	0.69	60.01	9.03	-
Sparse R-CNN [42]	94.12	52.58	0.67	59.65	8.45	-
SSD [13]	86.47	27.74	0.42	49.20	22.33	-
RetinaNet [43]	86.70	52.88	0.65	57.78	8.88	-
YOLOv3 [17]	84.63	42.71	0.54	50.48	20.33	-
YOLOv4 [18]	76.16	52.50	0.59	51.69	16.55	243.94
Zhang et al. [44]	-	-	-	86.10	178.57	483.33
Bai et al. [45]	-	-	0.71	68.1	70	-
Bai et al. [45]	-	-	0.72	69.8	66	-
YOLOv7-tiny [46]	89.8	74.9	0.81	80.9	-	6.2
YOLOv5s	91.48	67.96	0.78	76.39	112	7.03
Ours	92.50	74.97	0.83	81.00	86	6.90

TABLE 4. Detection results of CCTSDB 2021 in different weather conditions (unit: %).

Method	Sunny		Snow		Cloud		Night		Foggy	
	P	R	P	R	P	R	P	R	P	R
Faster R-CNN [11]	85.47	77.42	96.27	91.12	92.74	57.61	76.89	47.87	77.00	67.09
Libra R-CNN [40]	82.08	78.93	90.39	91.12	94.47	58.84	80.07	52.52	69.24	71.74
Dynamic R-CNN [41]	86.26	78.92	96.25	89.48	93.87	58.40	83.70	52.26	70.57	69.52
Sparse R-CNN [42]	96.56	73.27	95.01	88.49	96.72	55.97	91.48	44.15	92.11	81.11
SSD [13]	90.56	32.65	95.65	28.10	84.45	21.77	85.22	24.59	85.42	32.99
RetinaNet [43]	90.71	75.37	90.18	88.49	93.43	53.92	81.09	43.81	69.45	64.86
YOLOv3 [17]	92.01	64.03	87.54	70.59	87.12	44.65	75.98	34.81	88.66	56.39
YOLOv4 [18]	83.83	53.95	64.32	40.84	74.24	52.92	67.65	32.47	85.00	37.43
YOLOv5s	92.51	88.80	92.17	72.51	94.97	77.97	86.97	58.36	84.52	53.89
Ours	93.40	91.53	94.46	80.36	96.23	86.57	88.44	68.71	92.93	69.72

lighting during the night, which can lead to visual blurring and certain obstructions.

In a nighttime scene, the recent two-stage model, Sparse R-CNN, achieves a precision value (P) of 91.48% and a recall value (R) of 44.15%. On the other hand, the one-stage model YOLOv5s achieves a precision value of 86.97% and a recall value of 58.36%. Compared to YOLOv5s, Sign-YOLO demonstrates a 1.47% higher precision value and a 10.35% higher recall value in nighttime conditions. Under foggy weather conditions, Sign-YOLO shows significant improvements in both precision and recall, with P-values and R-values higher than YOLOv5s by 8.41% and 15.83%, respectively. Therefore, the proposed algorithm in this paper exhibits outstanding performance across a range of weather conditions, leading to a compelling conclusion.

This paper performed relevant tests in different road environments under natural scenes and varying weather conditions. The YOLOv5s algorithm, which demonstrated better performance, was selected for comparison with the Sign-YOLO algorithm proposed in this paper. The test results are shown in Figures 13–17. In these figures, 'prohibitory' represents prohibitory traffic signs, 'warning' represents warning traffic signs, and 'mandatory' represents regulatory traffic signs. Tests were conducted on rural and urban streets under sunny weather conditions, as shown in Figure 12. Under snowy weather conditions, tests were performed on highways and town roads, as illustrated in Figure 13.

FIGURE 12. Performance comparison of YOLOv5s and Sign-YOLO under sunny conditions (a) Original image; (b) YOLOv5s test results; (c) Sign-YOLO test results.

FIGURE 13. Performance comparison of YOLOv5s and Sign-YOLO under snow conditions (a) Original image; (b) YOLOv5s test results; (c) Sign-YOLO test results.

Similarly, under cloudy weather conditions, tests were carried out in the wilderness and residential areas, as demonstrated

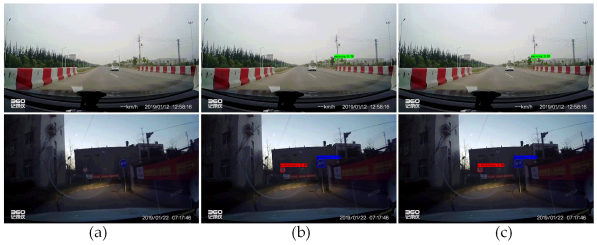

FIGURE 14. Performance comparison of YOLOv5s and Sign-YOLO under cloud conditions (a) Original image; (b) YOLOv5s test results; (c) Sign-YOLO test results.

FIGURE 15. Performance comparison of YOLOv5s and Sign-YOLO under night conditions (a) Original image; (b) YOLOv5s test results; (c) Sign-YOLO test results.

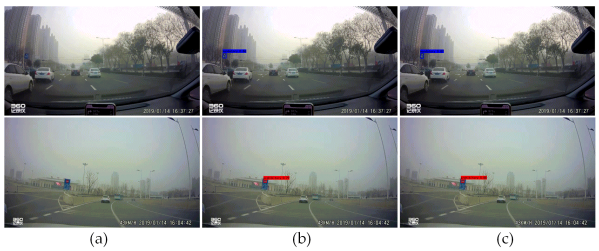

FIGURE 16. Performance comparison of YOLOv5s and Sign-YOLO under foggy conditions (a) Original image; (b) YOLOv5s test results; (c) Sign-YOLO test results.

TABLE 5. The detection performance comparison of different methods on the TT100K dataset.

Algorithm	mAP(%)	FPS	Size(M)
Faster R-CNN	73.4	3.0	143.7
YOLOv3-tiny [47]	44.15	60.4	34.3
YOLOv4-tiny [47]	46.34	87.9	23.4
Improved YOLOv4-tiny [47]	52.07	87.1	24.7
YOLOv7	-	-	37.4
YOLOv5s	75.04	102	7.04
Ours	78.89	87	6.90

in Figure 14. During nighttime weather conditions, tests were conducted on urban roads, as depicted in Figure 15. Additionally, under foggy weather conditions, tests were performed on urban roads, as shown in Figure 16. The results indicate that the algorithm presented in this paper achieves superior object localization and recognition performance compared to YOLOv5s. It can effectively meet the detection requirements of traffic signs for intelligent vehicles in various road environments within natural scenes.

In order to comprehensively validate the detection capabilities of our algorithm for Chinese traffic signs, we conducted a comparative analysis with different algorithms on the TT100K public dataset. The comparative results are shown in table 5. It is evident from the table that, compared with other mainstream algorithms, our algorithm demonstrates a significant advantage in the detection of Chinese traffic signs.

V. CONCLUSION

This paper proposes several improvement strategies based on YOLOv5s for traffic sign detection. Firstly, we used the K-means clustering approach to generate appropriate anchor boxes for the CCTSDB2021 traffic sign dataset. Next, we enhance the backbone network of YOLOv5s by incorporating a CA module into the CSP structure. This enhancement aims to boost the model's capability to extract crucial feature information, allowing the backbone network to focus on vital elements within traffic sign images while reducing background interference. Furthermore, we improve the Feature Pyramid Network (FPN) structure to better integrate local and global features. In the original FPN, features with different resolutions contribute unequally to the fused output features. Our proposed improvement addresses this issue by enhancing the network's ability to capture relevant information from both local and global features. CSP2_4 incorporates an advanced High-Ghost substitution technique instead of the traditional convolution operation, resulting in a reduction in parameter size for the Sign-YOLO model compared to YOLOv5s. The experimental results show that Sign-YOLO outperforms existing algorithms for traffic sign detection. Although the current method efficiently balances detection speed and accuracy on the CCTSDB2021 dataset, there is still an opportunity for improvement in detection speed.

REFERENCES

[1] A. Campbell, A. Both, and Q. Sun, "Detecting and mapping traffic signs from Google street view images using deep learning and GIS," *Comput., Environ. Urban Syst.*, vol. 77, Sep. 2019, Art. no. 101350.

[2] J. Zhang, X. Zou, L.-D. Kuang, J. Wang, R. S. Sherratt, and X. Yu, "CCTSDB 2021: A more comprehensive traffic sign detection benchmark," *Hum.-Centric Comput. Inf. Sci.*, vol. 12, pp. 1–21, May 2022.

[3] Y. Saadna, A. Behloul, and S. Mezzoudj, "Speed limit sign detection and recognition system using SVM and MNIST datasets," *Neural Comput. Appl.*, vol. 31, no. 9, pp. 5005–5015, Sep. 2019.

[4] R. P. Kumar, M. Sangeeth, K. S. Vaidhyanathan, and M. A. Pandian, "Traffic sign and drowsiness detection using open-CV," *TRAFFIC*, vol. 6, no. 3, pp. 1398–1401, 2019.

[5] A. Bouti, M. A. Mahraz, J. Riffi, and H. Tairi, "A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network," *Soft Comput.*, vol. 24, no. 9, pp. 6721–6733, May 2020.

[6] Z. Liang, J. Shao, D. Zhang, and L. Gao, "Traffic sign detection and recognition based on pyramidal convolutional networks," *Neural Comput. Appl.*, vol. 32, no. 11, pp. 6533–6543, Jun. 2020.

[7] Q. Yu and Y. Zhou, "Traffic safety analysis on mixed traffic flows at signalized intersection based on Haar-AdaBoost algorithm and machine learning," *Saf. Sci.*, vol. 120, pp. 248–253, Dec. 2019.

[8] A. Jayaprakash and C. KeziSelvaVijila, "Feature selection using ant colony optimization (ACO) and road sign detection and recognition (RSDR) system," *Cognit. Syst. Res.*, vol. 58, pp. 123–133, Dec. 2019.

[9] 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.

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

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Georgia, Oct. 2017, pp. 2980–2988.

[13] 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*. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.

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

[15] 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.

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

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

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

[19] C.-Y. Wang, A. Bochkovskiy, and H. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13024–13033.

[20] *Ultralytics/YOLOV5: V6.0*. Accessed: May 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[21] X. Dai, X. Yuan, G. Le, and L. Zhang, "Detection method of traffic signs based on color pair and MSER in the complex environment," *J. Beijing Jiaotong Univ.*, vol. 42, pp. 107–115, Jan. 2018.

[22] M.-J. Liang, X.-Y. Cui, Q.-S. Song, and X. Zhao, "Traffic sign recognition method based on HOG-Gabor feature fusion and Softmax classifier," *J. Traffic Transp. Eng.*, vol. 17, no. 3, pp. 151–158, 2017.

[23] X. Xu, J. Jin, S. Zhang, L. Zhang, S. Pu, and Z. Chen, "Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry," *Future Gener. Comput. Syst.*, vol. 94, pp. 381–391, May 2019.

[24] X.-Y. Sun, S.-J. Tang, J.-B. Guo, and C.-Q. Yu, "Traffic sign detection method based on adaptive gamma correction," *Comput. Simul.*, vol. 37, no. 12, pp. 414–420, 2020.

[25] M. J. F. Calero, M. A. Sanchez, J. Vargas, and M. J. Ayala, "Ecuadorian regulatory traffic sign detection by using HOG features and ELM classifier," *IEEE Latin Amer. Trans.*, vol. 19, no. 4, pp. 634–642, Apr. 2021.

[26] J. Wang, X. Leng, Z. Sun, X. Zhang, and K. Ji, "Fast and accurate refocusing for moving ships in SAR imagery based on FrFT," *Remote Sens.*, vol. 15, no. 14, p. 3656, Jul. 2023.

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

[28] J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 975–984, Mar. 2019.

[29] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, "YOLO v4 for advanced traffic sign recognition with synthetic training data generated by various GAN," *IEEE Access*, vol. 9, pp. 97228–97242, 2021.

[30] R. Ayachi, M. Afif, Y. Said, and A. Ben Abdelali, "An edge implementation of a traffic sign detection system for advanced driver assistance systems," *Int. J. Intell. Robot. Appl.*, vol. 6, no. 2, pp. 207–215, Jun. 2022.

[31] G. Lu, X. He, Q. Wang, F. Shao, J. Wang, and C. Hu, "A traffic sign detection network based on PosNeg-balanced anchors and domain adaptation," *Arabian J. Sci. Eng.*, vol. 48, no. 2, pp. 1333–1347, Feb. 2023.

[32] T. Liang, H. Bao, W. Pan, and F. Pan, "Traffic sign detection via improved sparse R-CNN for autonomous vehicles," *J. Adv. Transp.*, vol. 2022, pp. 1–16, Mar. 2022.

[33] T. Liang, H. Bao, W. Pan, X. Fan, and H. Li, "DetectFormer: Category-assisted transformer for traffic scene object detection," *Sensors*, vol. 22, no. 13, p. 4833, Jun. 2022.

[34] J. Park, S. Woo, J.-Y. Lee, and I. So Kweon, "BAM: Bottleneck attention module," 2018, *arXiv:1807.06514*.

[35] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[36] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[37] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.

[38] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[39] K. Han, Y. Wang, C. Xu, J. Guo, C. Xu, E. Wu, and Q. Tian, "GhostNets on heterogeneous devices via cheap operations," *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 1050–1069, Apr. 2022.

[40] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 821–830.

[41] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, "Dynamic R-CNN: Towards high quality object detection via dynamic training," in *Computer Vision—ECCV*: Glasgow, U.K.: Springer, 2020, pp. 260–275.

[42] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14449–14458.

[43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[44] J. Zhang, Z. Ye, X. Jin, J. Wang, and J. Zhang, "Real-time traffic sign detection based on multiscale attention and spatial information aggregator," *J. Real-Time Image Process.*, vol. 19, no. 6, pp. 1155–1167, Dec. 2022.

[45] W. Bai, J. Zhao, C. Dai, H. Zhang, L. Zhao, Z. Ji, and I. Ganchev, "Two novel models for traffic sign detection based on YOLOv5s," *Axioms*, vol. 12, no. 2, p. 160, Feb. 2023.

[46] 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 *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[47] L. Wang, K. Zhou, A. Chu, G. Wang, and L. Wang, "An improved lightweight traffic sign recognition algorithm based on YOLOv4-tiny," *IEEE Access*, vol. 9, pp. 124963–124971, 2021.

WEIZHEN SONG received the B.Eng. degree from Chongqing Three Gorges University, Chongqing, China, in 2015, and the M.Eng. degree from the Chongqing University of Posts and Telecommunications, Chongqing, in 2018. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, Universiti Sains Malaysia. His research interests include the perception technology of intelligent driving and deep learning.

SHAHREL AZMIN SUANDI (Senior Member, IEEE) received the B.Eng., M.Eng., and D.Eng. degrees from the Kyushu Institute of Technology, Fukuoka, Japan, in 1995, 2003, and 2006, respectively. He is currently a Professor and the Dean of the Electrical and Electronic Engineering School, Universiti Sains Malaysia, Malaysia. His research interests include object detection and tracking, and intelligent video surveillance.