frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 24 November 2023
DOI 10.3389/fphy.2023.1297828

Check for updates

OPEN ACCESS

EDITED BY
Chun-Hui He,
Xi'an University of Architecture and Technology, China

REVIEWED BY
Zhekang Dong,
Hangzhou Dianzi University, China
Bo Huang,
Shanghai University of Engineering Sciences, China

*CORRESPONDENCE
Xiaohong Zhang,
✉ xh.zhang@hpu.edu.cn

RECEIVED 20 September 2023
ACCEPTED 13 November 2023
PUBLISHED 24 November 2023

CITATION
Shen J, Zhang Z, Luo J and Zhang X (2023), YOLOv5-TS: Detecting traffic signs in real-time.
*Front. Phys.* 11:1297828.
doi: 10.3389/fphy.2023.1297828

COPYRIGHT
© 2023 Shen, Zhang, Luo and Zhang. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# YOLOv5-TS: Detecting traffic signs in real-time

Jiquan Shen[1,2], Ziyang Zhang[1,3], Junwei Luo[1] and Xiaohong Zhang[1]*

[1]School of Software, Henan Polytechnic University, Jiaozuo, China, [2]Anyang Institute of Technology, Anyang, China, [3]School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, China

Traffic sign detection plays a vital role in assisted driving and automatic driving. YOLOv5, as a one-stage object detection solution, is very suitable for Traffic sign detection. However, it suffers from the problem of false detection and missed detection of small objects. To address this issue, we have made improvements to YOLOv5 and subsequently introduced YOLOv5-TS in this work. In YOLOv5-TS, a spatial pyramid with depth-wise convolution is proposed by replacing maximum pooling operations in spatial pyramid pooling with depth-wise convolutions. It is applied to the backbone to extract multi-scale features at the same time prevent feature loss. A Multiple Feature Fusion module is proposed to fuse multi-scale feature maps multiple times with the purpose of enhancing both the semantic expression ability and the detail expression ability of feature maps. To improve the accuracy in detecting small even extra small objects, a specialized detection layer is introduced by utilizing the highest-resolution feature map. Besides, a new method based on k-means++ is proposed to generate stable anchor boxes. The experiments on the data set verify the usefulness and effectiveness of our work.

KEYWORDS

**multi-scale feature fusion, YOLOv5, object detection, traffic sign detection, k-means++**

## 1 Introduction

Traffic signs convey vital information such as speed limits, lane changes, pedestrian crossings, and potential hazards. With the ever-increasing volume of vehicles on the roads, accurately detecting and interpreting traffic signs is essential for assisting drivers in making informed decisions and complying with traffic regulations. It also plays a vital role in providing critical information to autonomous vehicles, allowing them to understand road regulations, make informed decisions, and navigate complex traffic scenarios.

Traditional approaches to traffic sign detection relied on color or shape template matching. However, these methods often struggle with variability in lighting conditions, shooting angles, and so on. In recent years, deep learning has shown significant advantages in object detection [1], attracting the attention of researchers. Several studies [2–4] have utilized Faster R-CNN [5] for traffic sign detection. R-CNN-based [6] methods detect objects in two stages, which limited detection speed, making them less suitable for real-time traffic sign detection scenarios. In contrast to R-CNN-based methods, You Only Look Once (YOLO) [7] can detect objects in one stage, offering faster detection speeds.

YOLOv5 is one of the variants of YOLO. It emphasizes both detection speed and accuracy. Therefore, it is very suitable for real-time traffic sign detection. However, it suffers from the false detection and miss detection of small even extra small objects. To improve the performance of YOLOv5 in detecting small even extra small objects Zhang et al.[8], introduced a new layer for detecting small objects, while [9,10] separately tried to reduce feature loss and alleviate the impact of feature loss in the process of feature extraction. Although many efforts have been devoted to improving the detection speed and detection accuracy, how to improve the performance in detecting small even extra small objects is still an open problem.

In this paper, we make several improvements to YOLOv5 and propose YOLOv5-TS to improve the performance in detecting small even extra small objects. We propose a spatial pyramid with Depth-wise convolution (SPDC) and combine it with a group of parallel strip convolution blocks to construct a multiple multi-scale feature fusion module (MFFM). MFFM is designed to extract multi-scale feature maps. Based on MFFM, we construct a special detection layer for small even extra small objects. We also improved the method to generate anchor boxes by exploiting k-means++ algorithm.

The main contributions of this work are described as follows:

(1) A spatial pyramid with depth-wise convolution is designed to extract multi-scale features without feature loss.
(2) A multiple feature fusion module is proposed to fuse multi-scale feature maps, and enhance the semantic and detailed expression capabilities.
(3) A new detection layer is constructed specially for detecting small even extra small objects.
(4) A new method based on the k-means++ algorithm is proposed to generate stable anchor boxes.

The rest of the paper is organized as follows. Section 2 provides an overview of related work in the field of traffic sign detection. Section 3 describes the theoretical basis. Section 4 details our proposed framework for traffic sign detection. Section 5 presents experimental results and performance evaluation. Finally, Section 6 concludes the paper and discusses avenues for future research in this domain.

## 2 Related work

Traditional traffic sign detection methods identify traffic signs by matching predefined color [11–13] or shape [14,15] templates. These methods are sensitive to lighting conditions and shooting angles, making it difficult to achieve stable detection results. Additionally, these methods detect traffic signs at low speeds and hence cannot work in real-time scenarios [16].

Deep learning has shown distinct advantages since its emergence [17–19]. It has been utilized to detect traffic signs [20,21]. Some researchers [2–4] detect traffic signs with R-CNN. However, R-CNN belongs to the category of two-stage object detection solutions. Although it can detect objects with high accuracy, it suffers from low detection speed. Therefore, it is not suitable for real-time traffic sign detection scenarios.

Different from R-CNN, You Look Only Once (YOLO) algorithm belongs to the category of one-stage detection solutions. It can detect objects at a high speed. YOLO has several versions, and some of the versions have been applied to traffic sign detection [22–24]. YOLOv5 is the version which emphasizes both detection accuracy and detection speed. Therefore, it is more suitable for real-time traffic sign detection than other solutions. Many efforts have been devoted to improve the performance of YOLOv5 in detecting traffic signs. To improve the detection speed of YOLOv5, Li et al.[25] used ghost convolution [26], depth-wise convolution [27] and channel attention [28] to construct a light version backbone. Zhao et al.[29] applied GSConv [30] to the feature fusion layer to reduce computation complexity. To improve the detection accuracy, Bai et al.[31] utilized a transformer structure to replace SPP. Wan et al.[32] improved the backbone with MixConv [33] and the neck with integrated attentional feature fusion [34].

Considering the impact of detection delay on real-time decision-making, detection operations should be conducted at a relatively long distance from traffic signs. Therefore, the detection targets, that is, traffic signs, are relatively small. However, YOLOv5 suffers from the false detection problem and the missed detection problem of small objects. To improve the accuracy of YOLOv5 in detecting small objects, Zhang et al.[8] constructed an additional detection for small objects. Mahaur and Mishra [9] replaced the pooling layers in the SPP module with dilated convolutions to capture the multi-scale features which is important for detecting small objects. Wang et al.[10] utilized an adaptive attention module and a features enhancement module to alleviate the loss of features of objects, especially small objects.

Although a lot of work has been devoted to improving the performance of YOLOv5 in detecting small objects, how to improve the accuracy of YOLOv5 in detecting small even extra small traffic signs in real time is still an open problem.

## 3 Theoretical basis

### 3.1 YOLOv5

YOLOv5 has serial versions, that is, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. In this work, we exploit YOLOv5s to detect traffic signs since YOLOv5s strikes a remarkable balance between speed and accuracy. Figure 1 shows the structure of YOLOv5s. According to the figure, YOLOv5s consists of backbone, neck and head. The backbone is primarily composed of CBS and CSP [35]. It extracts features from input data. The neck consists of FPN [36] and PAN [37]. It aims to enrich the features in each feature map. The head performs regression predictions according to the feature maps output from the neck.

### 3.2 SPPF

In YOLOv5, SPPF (Spatial Pyramid Pooling - Fast) is employed to capture information of different scales from an input feature map. It first utilizes a convolution layer to reduce the channels of the input image. And then, it exploits max pooling layer to generate feature maps of different scales. After that, it concatenates these feature maps along channels. Finally, it processes the concatenated feature map with a convolution operation to generate a feature map with rich features.

### 3.3 The k-means++ algorithm

The k-means++ clustering algorithm is an improved version of the k-means clustering algorithm. It is designed to optimize the selection of initial cluster centers. The k-means algorithm selects all initial cluster centers randomly, which may lead to local optimum. Different from k-means algorithm, k-means++ algorithm selects

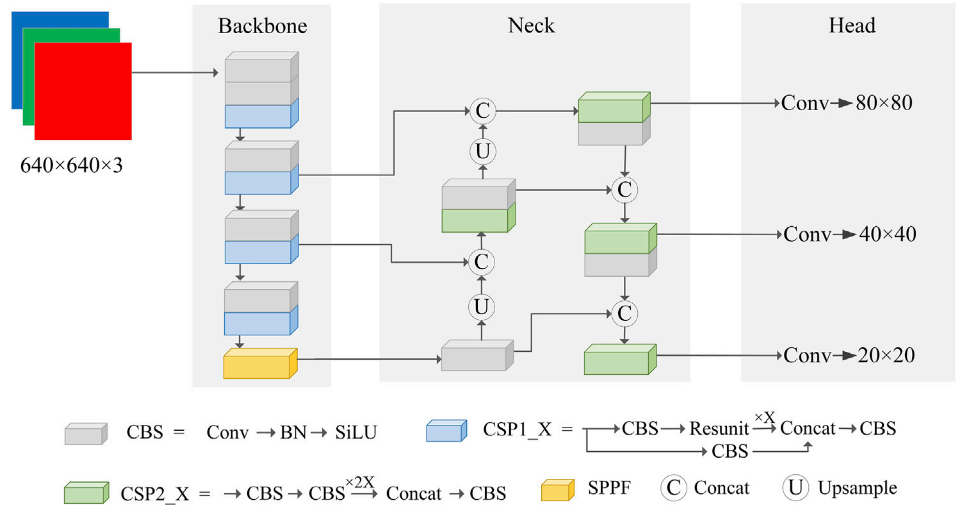

FIGURE 1
Structure of YOLOv5.

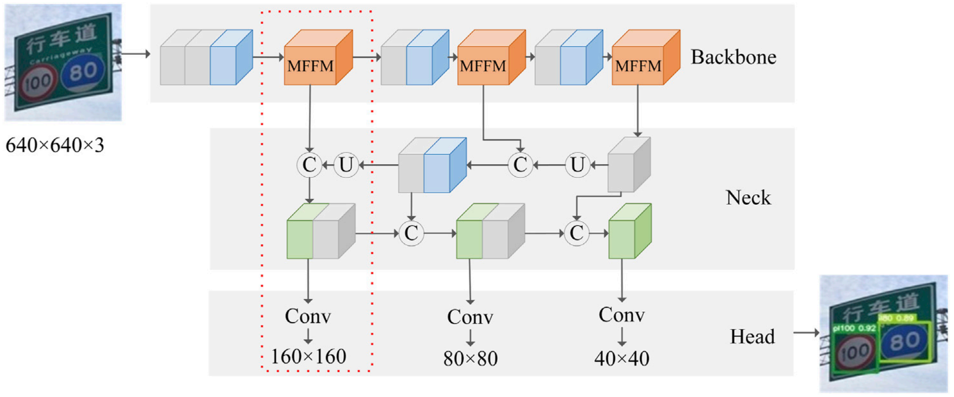

FIGURE 2
Structure of YOLOv5-TS. The modules circled by the red dotted rectangle construct the new detection layer. The orange cuboid represents the proposed MFFM.

only the first initial cluster centers randomly. After that, it selects the other initial cluster centers according to the distances to existing cluster centers. The k-means++ algorithm can not only converge more fastly, but also avoid trapping local optimum.

## 4 Proposed method

In order to improve the performance of YOLOv5 in detecting traffic signs, we improve YOLOv5 and propose YOLOv5-TS. In this section, we introduce the details of the improvements.

### 4.1 YOLOv5-TS

YOLOv5-TS is proposed based on the following improvements on YOLOv5. First, we propose SPDC by combining SPPF and depth-wise convolution. SPDC utilizes depth-wise convolutions to replace maximum pooling operations, thus avoiding the feature loss caused by the latter. It stacks multiple depth-wise convolutions to extract the multi-scale features of objects, which helps to capture the overall structures and local details of objects, and finally strengthens the expression ability of the fused feature map. Second, we propose a multiple multi-scale feature fusion module (MFFM) based on SPDC and a group of parallel strip convolution blocks. MFFM utilizes SPDC and the group of parallel strip convolution blocks to extract multi-scale feature maps, and exploits convolution and matrix operations to fuse those feature maps so as to enhance the semantic and detailed expression capabilities of feature maps. Third, we introduce a new detection layer of 160 × 160 to improve the performance in detecting extra small objects. Besides, we delete the layer to detect large objects since those objects are not common in traffic design scenarios. Finally, we optimize the method to generate anchor boxes by replacing the k-means algorithm with the k-means++ algorithm. Figure 2 shows the structure of YOLOv5-TS.

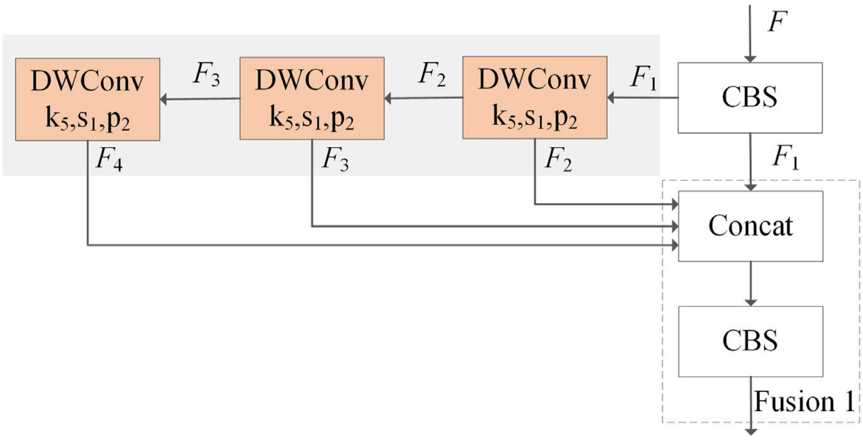

FIGURE 3
Structure of SPDC. DWConv expresses a depth-wise convolution, while $k_a$, $s_b$, $p_c$ indicate that the kernel size, stride, and padding of the depth-wise convolution are *a*, *b*, *c*, respectively.

## 4.2 Spatial pyramid involving depth-wise convolution

The Convolutional Neural Network (CNN) processes images only at specific scales. In reality, the scale of images is arbitrary. These images must be crapped or warped to a specific scale before being fed to CNNs [38]. However, crapping results in content loss and warping causes geometric distortion, which degrades detection accuracy. Spatial Pyramid Pooling (SPP) eliminates the limitation of deep convolutional neural network on the scale of input images by using multi-scale pooling, thereby avoiding the loss of features caused by cropping and the distortion caused by warping, and improving the accuracy of detection. Compared with SPP, Spatial Pyramid Pooling (SPPF) acquires feature maps of different receptive fields by stacking pooled layers with smaller kernels, and detecting objects with a higher speed.

SPP and SPPF utilize maximum pooling operations to extract features. The maximum pool operation retains only the maximum value in each region and discards all other values in the same region, which can lose critical information of targets. Compared with the maximum pooling operation, trained depth-wise convolution is more sensitive to the features of objects. It has the ability to retain the critical features of targets, which helps to improve the accuracy of detection. Based on the above analysis, we proposed a spatial pyramid with depth-wise convolution (SPDC). SPDC utilizes depth-wise convolutions to replace the multiple maximum pooling operations, thus avoiding the feature loss caused by the latter. It stacks multiple depth-wise convolutions to extract the multi-scale features of objects, which helps to capture the overall structures and local details of objects, and finally strengthens the expression ability of the fused feature map.

Figure 3 shows the structure of SPDC. According to the map, SPDC first utilizes CBS to reduce the channels of an input feature map and then uses three tandem depth-wise convolutions to extract three feature maps of different scales. After that, it concatenates all the feature maps generated in the previous steps and utilizes CBS to generate the output feature map. Given an input feature map-*F*, $f_{SPDC}(F)$ is utilized to represent the corresponding output of SPDC, and calculated according to Equation (1). In the equation, $f_{CBS}(\cdot)$ and $f_{con}$ $(\cdot)$ separately describe the functions related to CBS and concat. $F_1$, $F_2$, $F_3$, and $F_4$ describe the feature maps extracted by the first CBS and the three tandem depth-wise convolutions, respectively. They are calculated according to Eq. (2) ~ (5). In those equations, $f_D^{5\times5}$ denotes the depth-wise convolution with a kernel size of 5.

$$f_{SPDC}(F) = f_{CBS}\left(f_{con}(F_1, F_2, F_3, F_4)\right) \tag{1}$$

$$F_4 = f_D^{5\times5}(F_3) \tag{2}$$

$$F_3 = f_D^{5\times5}(F_2) \tag{3}$$

$$F_2 = f_D^{5\times5}(F_1) \tag{4}$$

$$F_1 = f_{CBS}(F) \tag{5}$$

## 4.3 Multiple feature fusion module

Large-scale feature maps are generated in shallow networks. They often contain rich details, such as color, texture, etc. These details are conducive to capturing the subtle features and local structure of objects, which benefits classification. Small-scale feature maps are generated by deep networks. After passing through multiple convolution layers, they lose some details but obtain rich semantic information which is conducive to capturing the overall shapes and locations of objects. If the small-scale feature map and the large-scale feature map are fused, a feature map containing rich semantic information and rich detailed information will be produced, which is conducive to improving the detection accuracy and generalization ability of the model. According to the above analysis, a Multiple Feature Fusion module (MFFM) is proposed and applied to the backbone network of the improved YOLOv5.

MFFM includes three feature fusion operations. The first two feature fusion operations are designed to fuse multi-scale feature maps, while the last fusion operation is exploited to fuse the output feature maps of the former two fusion operations and the

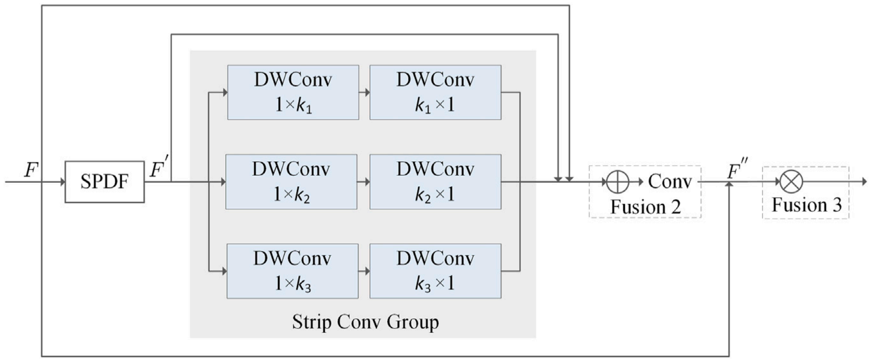

FIGURE 4
Structure of MFFM. ⊕ and ⊗ denote the matrix addition and matrix multiplication, respectively. Conv 1 × 1 describes 1 × 1 convolution. The operations surrounded by gray rectangle is utilized to extract multi-scale feature maps.

input feature map of MFFM to generate a new feature map with stronger feature expression ability. The first fusion operation is in SPDC. It is marked by the rectangle with dash lines in Figure 3. The multi-scale feature maps input to this fusion operation are extracted by the three tandem depth-wise convolutions in SPDC. The multi-scale feature maps input to the second fusion operation are extracted by a group of parallel strip convolution blocks, where each block is composed of two different strip convolutions. They are fused with the input feature map of MFFM and the output feature map of the first feature fusion operation. The third feature fusion is implemented by performing matrix multiplication on the input feature map of MFFM and the output feature map of the second feature fusion. Figure 4 describes the structure of MFFM.

$$f_S^i(F') = f_D^{k_i\times 1}\left(f_D^{1\times k_i}(F')\right) \tag{6}$$

$$f_{FU}^2(F', F) = f^{1\times 1}\left(f_{\oplus}\left(F, F', f_S^1(F'), f_S^2(F'), f_S^3(F')\right)\right) \tag{7}$$

$$f_{FU}^3(F'', F) = f_{\otimes}(F, F'') = f_{\otimes}\left(F, f_{FU}^2(F', F)\right) \tag{8}$$

Given an input feature map, $F$, the output of the first fusion operation is denoted as $f_{FU}^1(F)$. According to Figure 3, $f_{FU}^1(F)$ is equal to $f_{SPDC}(F)$, and can be calculated by Equation (1). Taking $F'$ to describe the input of the group of parallel strip convolution blocks, the output of the $i$th block can be denoted as $f_S^i(F')$ calculated according to Equation (6). It is used as the input of the second fusion operation together with the input of MFFM, the output of the first fusion operation, and the output of the group of parallel strip convolution blocks. Eq. (7) shows the calculation of the output of the second fusion operation. Based on the above equations, the output of the third fusion operation, that is, the output of MFFM, is calculated by Equation (8).

## 4.4 Multi-scale detection layers

YOLOv5 includes three detection layers. Those detection layers utilize three feature maps to detect objects of different scales, respectively. The first layer is constructed with the feature map of 80 × 80 in which each pixel can be mapped to an area of 8 × 8 in an input image. Therefore, it is suitable for detecting small objects. The second layer is constructed with the feature map of 40 × 40 of which each pixel corresponds to a region of 16 × 16 in an input image, and hence it is responsible for detecting medium objects. The third detection layer is utilized to detecting large objects since each pixel in the corresponding feature map, i.e., the feature map of 20 × 20, is related to an area of 32 × 32. However, considering the delay of detection and the real-time requirements of decision-making, detection should be carried out at a distance from objects, which indicates that the objects to be detected are usually small even extra small. Therefore, improving the detection performance of small even extra small objects is essential to improve the overall detection performance of traffic signs.

In order to improve the performance of traffic design, we introduce a special layer for extra small objects. The detection layer is constructed based on a 160 × 160 feature map extracted by the backbone network. Each pixel in the feature map corresponds to an area of 4 × 4 in an input image. The feature map is processed by MFFM to enhance the feature expression ability. After being processed by the neck network, it is used to predict extra small objects by the introduced layer. Considering that large objects are relatively not common in traffic sign detection scenarios and the feature map corresponding to detect large objects contains noise, we delete the detection layer of 20 × 20. Finally, the improved solution includes three detection layers used to detect extra small objects, small objects, and medium objects, respectively.

## 4.5 Anchor box generation with k-means++

Object detection algorithm always defines some bounding boxes in advance as anchor boxes. They set up multiple anchor boxes at each point, generate multiple prediction boxes according to these anchor boxes, and finally filter out qualified prediction boxes as detection results by indicators such as confidence. It is obvious that the selection of anchor boxes has a direct impact on detection results.

**TABLE 1 Anchor boxes clustering results.**

| Detection layers | Anchor boxes | |
|---|---|---|
| | Anchor boxes (k-means algorithm) | Anchor boxes (k-means++ algorithm) |
| (160 × 160) | (5, 6), (6, 7), (8, 9) | (5, 6), (7, 8), (9, 10) |
| (80 × 80) | (10, 11), (14, 14), (18, 19) | (12, 13), (16, 17), (20, 22) |
| (40 × 40) | (24, 26), (33, 34), (61, 55) | (27, 29), (34, 36), (49, 54) |

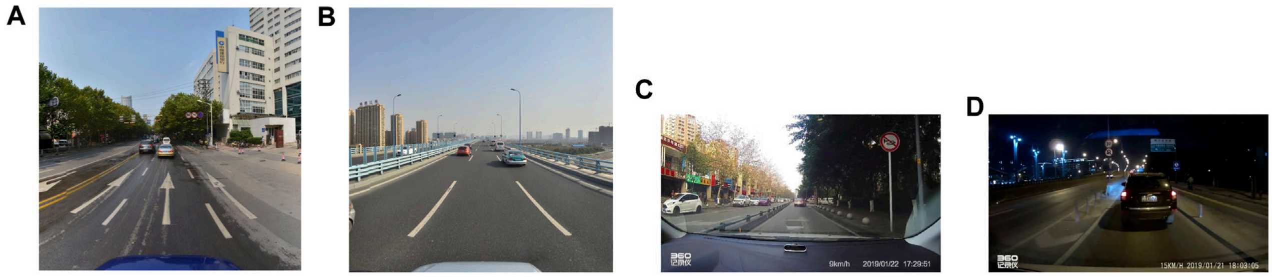

FIGURE 5
Image samples from TT100K-23 **(A)** and **(B)**, and CCTSDB2021 **(C)** and **(D)**.

YOLOv5 exploits the k-means algorithm to select anchor boxes. However, the k-means algorithm initializes the center points of k clusters in a random way, which can probably result in unstable clustering results. Different from the k-means algorithm, the k-means++ algorithm initializes the center point of only one cluster in a random way. It initializes center points for the remaining (k-1) clusters according to the shortest distances from each non-center point to all center points, which alleviates the instability caused by random policies. The clustering results of the two algorithms on TT100K-23 dataset are shown in Table 1.

# 5 Experiments

In this section, we present a detailed evaluation of YOLOv5-TS. First, we describe the experimental environments and evaluation metrics. Then, we describe the datasets used in this work. Finally, we discuss the results of the ablation experiments and the comparison experiments.

## 5.1 Experimental setup

**Environments.** All experiments are conducted on the same server equipped with an Intel Xeon Platinum 8260 Processor@ 2.30GHz, an NVIDIA RTX 3090 GPU, and 376 GB of memory. The server is deployed with Ubuntu 20.04.4, Torch 1.12.1, CUDA 11.3, and Python 3.8 development environment.

**Evaluation metrics.** We utilize five metrics, namely, Precision (P), Recall (R), mean Average Precision (mAP), F1-score, and Frames Per Second (FPS), to evaluate the performance of YOLOv5-TS. The first four metrics are calculated based on Eq. (9) ~ (13) where TP, FP, and FN represent the true positive samples, the false positive samples, and the false negative samples, respectively. In Equation (11), AP represents the average precision, P(R) describes the precision when recall is R. They are employed to assess the detection accuracy of YOLOv5-TS. The latter metric, FPS, is used to evaluate detection speed.

$$P = \frac{TP}{TP + FP} \tag{9}$$

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{\sum_{i=1}^{C} AP_i}{C} \tag{12}$$

$$F1 - score = \frac{2 \times P \times R}{P + R} \tag{13}$$

## 5.2 Dataset

We use two datasets, that is, TT100K-23 and CCTSDB2021, to evaluate our solution. The dataset samples are as shown in Figure 5.

**TT100K-23**: TT100K [39] comprises a total of 100,000 images, with only 10,000 images having been labeled. These labeled images contain a variety of 30,000 traffic signs, distributed among approximately 200 different classes. To create TT100K-23, we carefully chose 6,229 images, covering the 23 categories with the highest number of instances. TT100K is randomly divided into a training set and a test set in the ratio of 9:1.

**CCTSDB2021**: This dataset [40] contains 17,856 images with a total of 27,072 traffic signs. It is also divided into a training set and a test set in the ratio of 9:1. These traffic signs are divided into three classes: prohibitory, mandatory, and warning.

**TABLE 2** Ablation results.

| Models | MFFM | Detection layer | k-means++ | P (%) | R (%) | mAP (%) | F1-score |
|---|---|---|---|---|---|---|---|
| Model0 | × | × | × | 83.8 | 87.5 | 90.3 | 85.6 |
| Model1 | ✓ | × | × | 91.4 | 84.5 | 91.5 | 87.8 |
| Model2 | × | ✓ | × | 89.7 | 85.9 | 91.6 | 87.7 |
| Model3 | × | × | ✓ | 89.1 | 86.7 | 91.6 | 87.8 |
| Model4 | ✓ | ✓ | × | 91.3 | 87.1 | 92.6 | 89.1 |
| Model5 | × | ✓ | ✓ | 92 | 85.9 | 92.6 | 89 |
| Model6 | ✓ | × | ✓ | 91.1 | 87.2 | 92.5 | 89.1 |
| Model7 | ✓ | ✓ | ✓ | 92.5 | 86.8 | 93.7 | 89.5 |

**TABLE 3** Detection accuracy comparison on TT100K-23 dataset.

| Models | P (%) | R (%) | mAP (%) | F1-score |
|---|---|---|---|---|
| Faster-RCNN | 61.1 | 60.7 | 62.3 | 60.8 |
| RetinaNet | 45.4 | 59.9 | 44.9 | 51.6 |
| CenterNet | 82.4 | 66 | 71.8 | 73.2 |
| SSD | 97.3 | 24.2 | 85.8 | 38.7 |
| YOLOv3 | 90.1 | 78.6 | 85.3 | 83.9 |
| YOLOv4-tiny | 79.2 | 78 | 79.8 | 78.4 |
| YOLOv5n | 88.1 | 81.4 | 88.2 | 84.6 |
| YOLOv5s | 83.8 | 87.5 | 90.3 | 85.6 |
| YOLOv5m | 88.4 | 86.4 | 90.6 | 87.3 |
| YOLOX | 89.7 | 85.8 | 89.2 | 87.7 |
| YOLOv7 | 93.2 | 88.1 | 94.4 | 90.5 |
| YOLOv8n | 90 | 77.8 | 89.2 | 83.4 |
| YOLOv8s | 89.2 | 80.9 | 90.9 | 84.8 |
| YOLOv5-TS | 92.5 | 86.8 | 93.7 | 89.5 |
| solution[25] | 90.9 | 85.2 | 91.3 | 87.9 |
| solution[51] | 86.4 | 87.4 | 92 | 86.8 |
| solution[50] | 91.5 | 84.3 | 91.5 | 87.7 |
| solution[9] | 87.5 | 86 | 90.3 | 86.7 |

**TABLE 4** Detection accuracy comparison on CCTSDB2021 dataset.

| Models | P (%) | R (%) | mAP (%) | F1-score |
|---|---|---|---|---|
| Faster-RCNN | 95.6 | 65.2 | 95.6 | 77.5 |
| RetinaNet | 94.1 | 66.4 | 94 | 77.8 |
| CenterNet | 93.8 | 82.8 | 87.4 | 87.9 |
| SSD | 98.1 | 39.5 | 94.1 | 56.3 |
| YOLOv3 | 97.2 | 94.8 | 98 | 95.9 |
| YOLOv4-tiny | 91.7 | 89.1 | 92.9 | 90.3 |
| YOLOv5n | 97.4 | 94.2 | 97.8 | 95.7 |
| YOLOv5s | 96.1 | 95.7 | 97.9 | 95.8 |
| YOLOv5m | 97.8 | 97.8 | 98.8 | 97.8 |
| YOLOX | 96.7 | 97.6 | 98.3 | 97.1 |
| YOLOv7 | 94.8 | 96.4 | 97.1 | 95.5 |
| YOLOv8n | 97.1 | 96.3 | 98.5 | 96.6 |
| YOLOv8s | 97.3 | 96.8 | 98.8 | 97 |
| YOLOv5-TS | 97.6 | 98.4 | 99.1 | 98 |
| solution[25] | 97 | 97.2 | 98.6 | 97.1 |
| solution[51] | 96.3 | 97.5 | 98.9 | 96.8 |
| solution[50] | 97.5 | 97.6 | 98.9 | 97.5 |
| solution[9] | 96.9 | 96.6 | 98.5 | 96.7 |

## 5.3 Ablation study

To analyze the effectiveness of the improvements in this work, we perform ablation experiments on the TT100K-23 dataset and describe the results in Table 2. In this table, Model0 denotes the original version of YOLOv5s, while Model1, Model2, and Model3 denote the variant version with a new detection layer, the variant with MFFM, and the variant with the anchor box generation method based on the k-means++ algorithm, respectively. According to the experimental results, Model1, Model2, and Model3 all achieved performance improvement compared to Model0, indicating that the introduction of the new detection layer, MFFM, and the anchor box generation method based on k-means++ helped to improve the performance of YOLOv5s in detecting traffic signs. Model4 represents the variant version with MFFM and 160*160 detection layer. It obtained performance improvements compared to Model0. In addition, in the experiment, the values of $k_1$, $k_2$, and $k_3$ in Equation (6) are set to 7, 11, and 21, respectively.

## 5.4 Performance comparison

We conduct experiments to evaluate the detection accuracy and speed of YOLOv5-TS by comparing it to several important object

**TABLE 5 Detection speed comparison on TT100K-23 dataset.**

| Models | FPS(f/s) |
|---|---|
| Faster-RCNN | 22 |
| RetinaNet | 20 |
| CenterNet | 32 |
| SSD | 47 |
| YOLOv3 | 21 |
| YOLOv4-tiny | 51 |
| YOLOv5n | 75 |
| YOLOv5s | 70 |
| YOLOv5m | 54 |
| YOLOX | 25 |
| YOLOv7 | 23 |
| YOLOv8n | 71 |
| YOLOv8s | 63 |
| YOLOv5-TS | 67 |
| solution[25] | 65 |
| solution[51] | 65 |
| solution[50] | 64 |
| solution[9] | 69 |

**TABLE 6 Detection speed comparison on CCTSD2021 dataset.**

| Models | FPS(f/s) |
|---|---|
| Faster-RCNN | 25 |
| RetinaNet | 21 |
| CenterNet | 34 |
| SSD | 61 |
| YOLOv3 | 27 |
| YOLOv4-tiny | 65 |
| YOLOv5n | 77 |
| YOLOv5s | 74 |
| YOLOv5m | 59 |
| YOLOX | 25 |
| YOLOv7 | 26 |
| YOLOv8n | 73 |
| YOLOv8s | 63 |
| YOLOv5-TS | 71 |
| solution[25] | 64 |
| solution[51] | 67 |
| solution[50] | 64 |
| solution[9] | 57 |

detection solutions, including Faster-RCNN [5], RetinaNet [41], CenterNet [42], SSD [43], YOLOv3 [44], YOLOv4 [45], YOLOv5n, YOLOv5s, YOLOv5m [46], YOLOX [47], YOLOv7 [48], YOLOv8n, and YOLOv8s [49]. Faster-RCNN is a two-stage algorithm, while all the other algorithms are one-stage algorithms. To ensure the fairness of training processes, the training parameters of batch_size and the number of iterations are separately set to 32 and 800, while all the other training parameters are configured with their default values.

We use P, R, mAP, and F1-score to evaluate detection accuracy and record the corresponding results in Table 3 and Table 4.

Table 3 shows the detection results on TT100K-23 dataset. According to the results, YOLOv5-TS obtained the highest P, R, mAP, and F1-score compared with all the other variants of YOLO except YOLOv5s and YOLOv7. Although YOLOV5-TS had a lower R than YOLOv5s, it obtained a higher P, mAP and F1-score, which indicates that YOLOv5-TS performed better than YOLOv5s.

Table 4 shows the detection results on CCTSDB2021 dataset. According to the results, YOLOv5-TS obtained the highest P, R, mAP, and F1-score compared with all the other variants of YOLO except YOLOv5m. Although YOLOv5m outperformed YOLOv5-TS on the p metric, it is surpassed by YOLOv5-TS on the other three metrics. Therefore, we think YOLOv5-Ts performs better than YOLOv5m on CCTSDB2021.

The results on TT100K-23 show that YOLOv7 had an obvious advantage over YOLOv5-TS. However, the advantage was given away on CCTSDB2021 according to Table 4. To further evaluate YOLOv5-TS and YOLOV7, we carried out the experiments to evaluate the detection speeds of different solutions since traffic sign detection is predominantly applied in real-time scenarios which demand not only high detection accuracy but also swift detection speed. Table 5 and Table 6 show the corresponding results. According to the table, YOLOv5-TS processed 67 frames per second, whereas YOLOv7 only handled 23 frames per second. This suggests that YOLOv5-TS is significantly more well-suited for real-time traffic sign detection than YOLOv7.

SSD is a one-stage solution. According to the results in Table 3 and Table 4, SSD surpassed YOLOv5-TS in terms of P metric, but lagged behind YOLOv5-TS in terms of all the other three metrics. According to the results in Tables 5 and Table 6, SSD detected traffic signs at a speed much slower than YOLOv5-TS did. Considering the above results, we think YOLOv5-TS performs better than SSD. RetinaNet and CenterNet are also one-stage algorithms. According to the results in Tables 3–6, our solution performed better than them. The results in Tables 3–6 also indicated that our solution performed better than Faster-RCNN which is a two-stage algorithm.

To further evaluate YOLOv5-TS, we compared it with four different solutions, that is, [9,25,50,51]. All these solutions utilized YOLOv5 to detect traffic signs. They all made improvements to YOLOv5 and obtained performance gains. The corresponding results are recorded in Tables 3–6. According to these four tables, YOLOv5-TS outperformed better on TT100K dataset and CCTSDB2021 dataset than the four solutions, regardless of which of the four evaluation indicators was used.

Figure 6 shows the detected results of YOLOv5-Ts on the images captured at different distances, light conditions and shooting angles. According to the results, YOLOv5-TS correctly recognized all the small-size traffic signs in all the images.

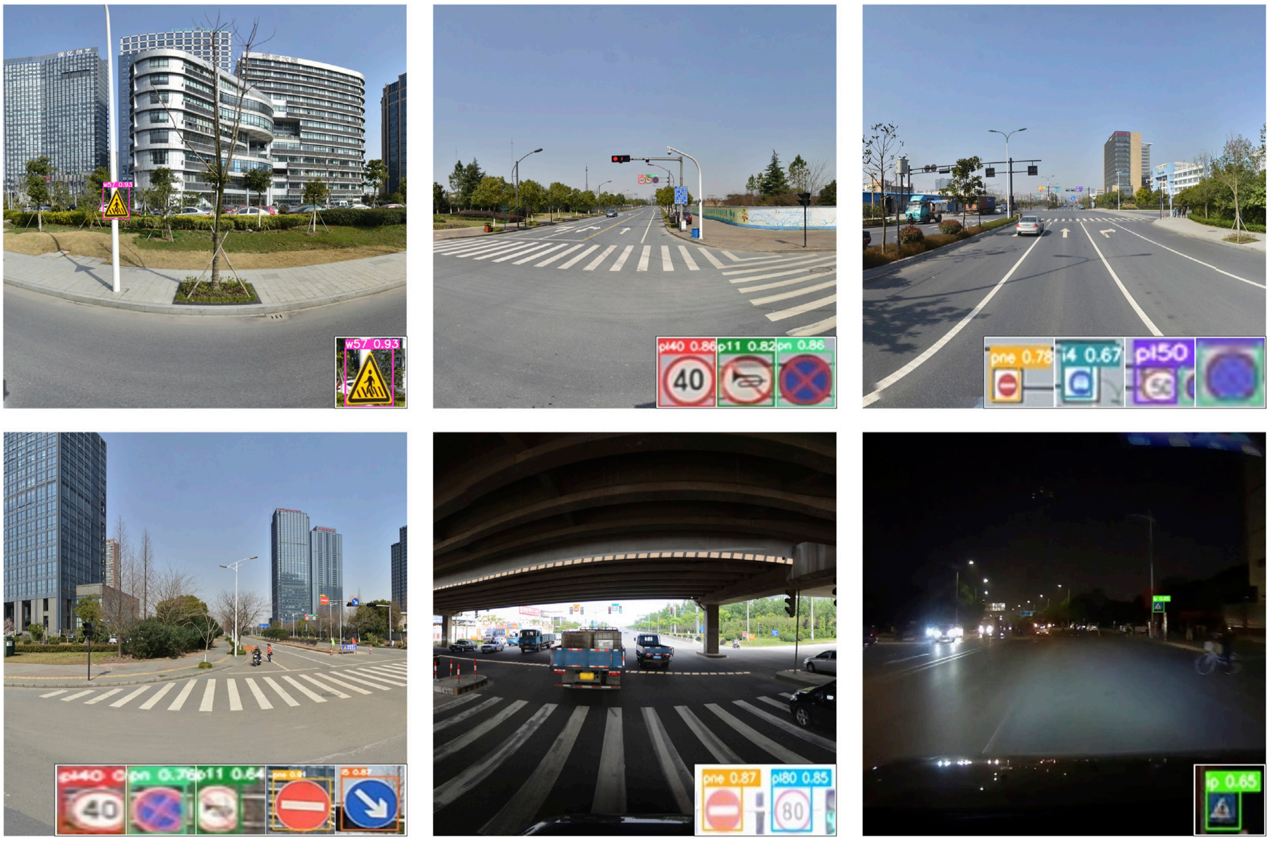

FIGURE 6
Detection performance of the YOLOv5-TS model trained on the TT100K dataset. The detection results of the target are magnified and displayed at the bottom of the image.

## 6 Conclusion

In this work, we analyzed the performance problem of YOLOv5 in real-time traffic sign detection. To address the performance issues, we proposed several enhancements to YOLOv5. Firstly, we introduced a spatial pyramid with depth-wise convolution to address feature loss in the SPPF module and extract multi-scale features more effectively. Secondly, we propose a multiple feature fusion module to further extract and fuse multi-scale features, enhancing feature representation. Thirdly, we introduced a specialized detection layer to improve the accuracy in detecting small even extra small traffic signs. Finally, we incorporated the k-means++ clustering algorithm to obtain anchor boxes better suited for the data sets. Experimental results demonstrate that the improved model effectively enhances accuracy without significantly increasing model complexity. In the future, we will implement the improvements in the work to YOLOv8.

## Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

## Author contributions

JS: Writing–original draft, Writing–review and editing. ZZ: Writing–original draft, Writing–review and editing. JL: Writing–original draft. XZ: Writing–original draft.

## Funding

The authors declare financial support was received for the research, authorship, and/or publication of this article. This work has been supported in part by the National Natural Science Foundation of China under Grant No. 61972134, Innovative and Scientific Research Team of Henan Polytechnic University under No. T2021-3.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

1. Tulbure A-A, Tulbure A-A, Dulf E-H. A review on modern defect detection models using dcnns–deep convolutional neural networks. *J Adv Res* (2022) 35:33–48. doi:10.1016/j.jare.2021.03.015

2. Li X, Xie Z, Deng X, Wu Y, Pi Y. Traffic sign detection based on improved faster r-cnn for autonomous driving. *The J Supercomputing* (2022) 78:7982–8002. doi:10.1007/s11227-021-04230-4

3. Han C, Gao G, Zhang Y. Real-time small traffic sign detection with revised faster-rcnn. *Multimedia Tools Appl* (2019) 78:13263–78. doi:10.1007/s11042-018-6428-0

4. Song Y, Fan R, Huang S, Zhu Z, Tong R. A three-stage real-time detector for traffic signs in large panoramas. *Comput Vis Media* (2019) 5:403–16. doi:10.1007/s41095-019-0152-1

5. Ren S, He K, Girshick R, Sun J. Faster r-cnn: towards real-time object detection with region proposal networks. *Adv Neural Inf Process Syst* (2015) 28.

6. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2014; Columbus, OH, USA (2014). p. 580–7.

7. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2016; Las Vegas, Nevada, USA (2016). p. 779–88.

8. Zhang S, Che S, Liu Z, Zhang X. A real-time and lightweight traffic sign detection method based on ghost-yolo. *Multimedia Tools Appl* (2023) 82:26063–87. doi:10.1007/s11042-023-14342-z

9. Mahaur B, Mishra K. Small-object detection based on yolov5 in autonomous driving systems. *Pattern Recognition Lett* (2023) 168:115–22. doi:10.1016/j.patrec.2023.03.009

10. Wang J, Chen Y, Dong Z, Gao M. Improved yolov5 network for real-time multi-scale traffic sign detection. *Neural Comput Appl* (2023) 35:7853–65. doi:10.1007/s00521-022-08077-5

11. Zhang K, Sheng Y, Li J. Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. *IET Intell Transport Syst* (2012) 6:282–91. doi:10.1049/iet-its.2011.0105

12. Gómez-Moreno H, Maldonado-Bascón S, Gil-Jiménez P, Lafuente-Arroyo S. Goal evaluation of segmentation algorithms for traffic sign recognition. *IEEE Trans Intell Transportation Syst* (2010) 11:917–30. doi:10.1109/tits.2010.2054084

13. Salti S, Petrelli A, Tombari F, Fioraio N, Di Stefano L. Traffic sign detection via interest region extraction. *Pattern Recognition* (2015) 48:1039–49. doi:10.1016/j.patcog.2014.05.017

14. Barnes N, Zelinsky A, Fletcher LS. Real-time speed sign detection using the radial symmetry detector. *IEEE Trans Intell Transportation Syst* (2008) 9:322–32. doi:10.1109/tits.2008.922935

15. Fang C-Y, Chen S-W, Fuh C-S. Road-sign detection and tracking. *IEEE Trans vehicular Technol* (2003) 52:1329–41. doi:10.1109/TVT.2003.810999

16. Abbas Q, Ibrahim ME, Jaffar MA. A comprehensive review of recent advances on deep vision systems. *Artif Intelligence Rev* (2019) 52:39–76. doi:10.1007/s10462-018-9633-3

17. Dong Z, Ji X, Zhou G, Gao M, Qi D. Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications. *IEEE Trans Industry Appl* (2022) 59:47–58. doi:10.1109/tia.2022.3188749

18. Dong Z, Lai CS, Zhang Z, Qi D, Gao M, Duan S. Neuromorphic extreme learning machines with bimodal memristive synapses. *Neurocomputing* (2021) 453:38–49. doi:10.1016/j.neucom.2021.04.049

19. Dong Z, Ji X, Lai CS, Qi D, Zhou G, Lai LL. Memristor-based hierarchical attention network for multimodal affective computing in mental health monitoring. *IEEE Consumer Elect Mag* (2022) 12:94–106. doi:10.1109/mce.2022.3159350

20. Wali SB, Abdullah MA, Hannan MA, Hussain A, Samad SA, Ker PJ, et al. Vision-based traffic sign detection and recognition systems: current trends and challenges. *Sensors* (2019) 19:2093. doi:10.3390/s19092093

21. Dong S, Wang P, Abbas K. A survey on deep learning and its applications. *Comp Sci Rev* (2021) 40:100379. doi:10.1016/j.cosrev.2021.100379

22. Yao Y, Han L, Du C, Xu X, Jiang X. Traffic sign detection algorithm based on improved yolov4-tiny. *Signal Processing: Image Commun* (2022) 107:116783. doi:10.1016/j.image.2022.116783

23. Zhang J, Ye Z, Jin X, Wang J, Zhang J. Real-time traffic sign detection based on multiscale attention and spatial information aggregator. *J Real-Time Image Process* (2022) 19:1155–67. doi:10.1007/s11554-022-01252-w

24. Li Y, Li J, Meng P. Attention-yolov4: a real-time and high-accurate traffic sign detection algorithm. *Multimedia Tools Appl* (2023) 82:7567–82. doi:10.1007/s11042-022-13251-x

25. Li W, Zhang G, Cui L. A novel lightweight traffic sign recognition model based on yolov5. *J transportation Eng A: Syst* (2023) 149:04023025. doi:10.1061/jtepbs.teeng-7461

26. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2020; Seattle, WA, USA (2020). p. 1580–9.

27. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). https://arxiv.org/abs/1704.04861.

28. Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2021; Nashville, TN, USA (2021). p. 13713–22.

29. Zhao L, Wei Z, Li Y, Jin J, Li X. Sedg-yolov5: a lightweight traffic sign detection model based on knowledge distillation. *Electronics* (2023) 12:305. doi:10.3390/electronics12020305

30. Li H, Li J, Wei H, Liu Z, Zhan Z, Ren Q. Slim-neck by gsconv: a better design paradigm of detector architectures for autonomous vehicles (2022). https://arxiv.org/abs/2206.02424.

31. Bai W, Zhao J, Dai C, Zhang H, Zhao L, Ji Z, et al. Two novel models for traffic sign detection based on yolov5s. *Axioms* (2023) 12:160. doi:10.3390/axioms12020160

32. Wan H, Gao L, Su M, You Q, Qu H, Sun Q. A novel neural network model for traffic sign detection and recognition under extreme conditions. *J Sensors* (2021) 2021:1–16. doi:10.1155/2021/9984787

33. Tan M, Le QV. Mixconv: mixed depthwise convolutional kernels (2019). https://arxiv.org/abs/1907.09595.

34. Dai Y, Gieseke F, Oehmcke S, Wu Y, Barnard K. Attentional feature fusion. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision; January 2021; Waikoloa, HI, USA (2021). p. 3560–9.

35. Wang C-Y, Liao H-YM, Wu Y-H, Chen P-Y, Hsieh J-W, Yeh I-H. 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; June 2020; Seattle, WA, USA (2020). p. 390–1.

36. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition; July, 2017; Honolulu, HI, USA (2017). p. 2117–25.

37. Liu S, Qi L, Qin H, Shi J, Jia J. Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2018; Salt Lake City, UT, USA (2018). p. 8759–68.

38. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE* (1998) 86:2278–324. doi:10.1109/5.726791

39. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S. Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2016; Las Vegas, NV, USA (2016). p. 2110–8.

40. Zhang J, Zou X, Kuang L-D, Wang J, Sherratt RS, Yu X. Cctsdb 2021: a more comprehensive traffic sign detection benchmark. *Human-centric Comput Inf Sci* (2022) 12. doi:10.22967/HCIS.2022.12.023

41. Lin T-Y, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. *IEEE Trans Pattern Anal Machine Intelligence* (2017) 2980–8. Available at: https://arxiv.org/abs/1708.02002 (Accessed February 7, 2018).

42. Zhou X, Wang D, Krähenbühl P. Objects as points (2019). https://arxiv.org/abs/1904.07850.

43. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al. Ssd: single shot multibox detector. In: Proceedings of the Computer Vision–ECCV 2016: 14th European Conference; October 2016; Amsterdam, The Netherlands. Springer (2016). p. 21–37.

44. Redmon J, Farhadi A. Yolov3: an incremental improvement (2018). https://arxiv.org/abs/1804.02767.

45. Bochkovskiy A, Wang C-Y, Liao H-YM. Yolov4: optimal speed and accuracy of object detection (2020). https://arxiv.org/abs/2004.10934.

46. Jocher G. *YOLOv5 by ultralytics* (2020).

47. Ge Z, Liu S, Wang F, Li Z, Sun J. Yolox: exceeding yolo series in 2021 (2021). https://arxiv.org/abs/2107.08430.

48. Wang C-Y, Bochkovskiy A, Liao H-YM. 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; June 2020; Vancouver, BC, Canada (2023). p. 7464–75.

49. Jocher G, Chaurasia A, Qiu J. *YOLO by ultralytics* (2023).

50. Dang TP, Tran NT, To VH, Tran Thi MK. Improved yolov5 for real-time traffic signs recognition in bad weather conditions. *J Supercomputing* (2023) 79:10706–24. doi:10.1007/s11227-023-05097-3

51. Han Y, Wang F, Wang W, Li X, Zhang J. Yolo-sg: small traffic signs detection method in complex scene. *J Supercomputing* (2023) 1–22. doi:10.1007/s11227-023-05547-y