Multimedia Tools and Applications (2023) 82:26063–26087
https://doi.org/10.1007/s11042-023-14342-z

A real-time and lightweight traffic sign detection method based on ghost-YOLO

Shuo Zhang [1] · Shengbing Che [1] · Zhen Liu [1] · Xu Zhang [1]

Received: 20 April 2022 / Revised: 11 November 2022 / Accepted: 2 January 2023 /
Published online: 14 January 2023

Abstract
Traffic sign detection is an essential part of traffic security and unmanned driving system. Due to the changes in the traffic environment is complex, how to intelligently and efficiently detect traffic signs in real scenes is of great significance. The traffic sign detection task is characterized by many small targets and complex environmental interference, and the detection scene also requires the detection model to be lightweight and efficient. This paper proposes a lightweight model Ghost-YOLO, and a lightweight module C3Ghost is designed to replace the feature extraction module in YOLOv5. C3Ghost modules extract features in a lightweight way, which effectively speeds up inference. At the same time, a new multi-scale feature extraction is designed to enhance the focus on small targets. Experimental results show that the mAP of the Ghost-YOLO is 92.71%, and the number of parameters and computations are respectively reduced to 91.4% and 50.29% of the original. Compared with multiple lightweight models, the speed and accuracy of this method are competitive.

Keywords Deep learning · Traffic sign detection · Small object detection · Ghost-YOLO

1 Introduction

As the foundation of the national economy, the road transportation system is developing rapidly. Meanwhile, traffic problems have become increasingly prominent, such as urban traffic jams, frequent traffic accidents, and increased air pollution. Furthermore, Traffic accidents endanger personal safety and social security. Analysis of the frequent causes of accidents mainly includes fatigue driving, illegal driving, bad weather, etc. Among them, driver's subjective driving behavior such as driving in violation of traffic signs is one of the main causes of traffic accidents. Therefore, it is necessary to develop an Intelligent Transport

✉ Shengbing Che
T20050514@csuft.edu.cn

[1] College of Computer and Information Engineering, Central South University of Forestry and Technology, Changsha 410004, China

System (ITS) to assist the driver [51]. In addition, with the global spread of the new crown epidemic in 2020, unmanned vehicles are used in many hospitals to distribute emergency supplies, making unmanned vehicles have once again entered the public's attention. To sum up, the Traffic Sign Detection (TSD) system is an important sub-module of ITS [48], and its detection accuracy is an important prerequisite for ITS to effectively assist the driver and the unmanned system to drive safely.

TSD can be regarded as a sub-task in the field of object detection, where the goal is to detect traffic signs and their boundaries. Traffic signs are designed in a specific pattern, differentiated from their surroundings mainly by color, shape, and what they signify. Therefore, early traffic sign recognition algorithms were mainly aimed at the localization and classification of target regions. Traditional methods [55] used color thresholding and shape analysis to segment traffic signs from images. With the development of computer vision technology, deep learning has demonstrated the powerful ability to learn feature representations from raw data, which has received great attention in pattern recognition and computer vision research. It has been widely used in object detection and recognition. For example, Convolutional Neural Networks (CNNs) have shown their powerful feature extraction ability [6]. Many CNN-based methods have achieved fruitful results in object detection tasks.

However, traffic sign recognition still faces the following challenges:

1. Under different viewing angles and viewing distances, the traffic sign images may appear distortions of shape and color.
2. The complicated road environment can lead to complex background of the traffic signs.
3. Traffic signs have characteristics that most object detection objects do not have, so it is hard to obtain satisfactory performance by simply applying conventional object detection methods.

In order to address the above challenges, scholars have done extensive studies. However, these original methods are difficult to be widely used in practical detection scenarios. On the one hand, designing these feature extraction methods for specific traffic sign categories requires a lot of work and consumes manpower and material resources. On the other hand, simple feature extraction methods are not powerful enough to deal with the complex and changing traffic environment. In addition, in the real traffic images which can be captured by in-vehicle equipment, traffic signs often occupy only a small part of the area as shown in Fig. 1. Conventional object detection classifiers used a series of down-sampling operations to obtain high-level feature maps. It will lead to the loss of small targets, which is unfavorable for the TSD task dominated by small targets. So it is difficult to obtain satisfactory performance simply by using traditional object detection methods. For this reason, a series of excellent basic networks such as VGGNet [39], ResNet [10] and DenseNet [14] have been proposed. The typical models also include Fast R-CNN [7], YOLO [32], SSD [23], and RetinaNet [20].

Although increasing the complexity of the detection classifier can improve the detection effect, the complexity of model heavily increases the number of parameters and computation. In actual scenarios, the TSD system should be deployed on the premise of onboard embedded devices to effectively identify traffic signs. Too large models are difficult to meet the real-time performance required by industrial applications, so quantitative networks came into being. The SqueezeNet [15] network uses common compression techniques to compress the model and then expand. On the basis of similar performance to AlexNet [18], the parameter model was only 1/50 of AlexNet. However, the network still adopted the standard convolution calculation

(a) (b)

Fig. 1 Image sample in TT100K. **a** Original images in TT100K dataset and the green rectangle regions contain traffic signs; **b** Image patches that are cropped from (a) according to the green rectangle

method. MobileNet [12] employed a more effective depthwise separable convolution, which improved the network speed and further promotes the application of the convolution network on the mobile terminal. Furthermore, higher precision is obtained with less computation. Theoretically, the amount of computation can still be reduced. ShuffleNet [52] used group convolution and channel shuffling to effectively reduce the amount of computation for point convolution, which achieved better performance. With the advancement of mobile devices and the diversified development of application scenarios, lightweight networks show higher engineering value. Therefore, how simultaneously ensuring the accuracy and speed of TSD is still a difficult problem.

Inspired by the above methods, the purpose of this study is to develop a lightweight method for TSD which can strike a balance between accuracy and efficiency and can solve the problem of small target loss. In this paper, a new traffic sign recognition method called Ghost-YOLO was proposed. The main contributions can be summarized as follows:

1. The detection and recognition of traffic signs in the actual environment is one of the technical bottlenecks of ITS. Through experimental research and test, this paper provided a scientific means and framework for accurate recognition of traffic signs.
2. The Ghost-YOLO model was proposed. Based on Ghost Conv, a more lightweight C3Ghost structure is proposed to replace the backbone network of the YOLOv5 target detection model. After realizing model compression and speeding up inference, this paper obtains an optimized neural network model, which greatly reduced the dependence on the hardware environment.
3. Aiming at the fact that there are many small objects in the actual scene of TSD, the multi-scale feature fusion detection head is used to detect large, medium and small scales, which improves the detection performance of small objects.
4. Experimental results on TT100k dataset show that compared with several current advanced object detection methods, this method can obtain a more lightweight model scale

on the basis of maintaining competitive performance, which enhances the practicability of the model

The rest of this paper is organized as follows. Section 2 introduces the related work of TSD in recent years. Section 3 introduces the C3Ghost module design method, multi-scale feature fusion scheme, and overall model structure. Section 4 describes the dataset, experimental setup and experimental results. Finally, Section 5 provides a summary and outlook of this paper.

2 Related work

2.1 Traditional TSD

Traffic signs are usually designed in a specific pattern, mainly distinguished by color, shape, and the content they mark, which are often different from the surroundings. Color and shape are the basic attributes of traffic signs and their information started to be used for identification in early research. The core of the algorithm for detecting based on color is to select the color space of the image, and the images collected by the in-vehicle equipment are generally RGB images. Benallal et al. found that the RGB components are significantly different under different lighting conditions. Segmenting the RGB images collected by the camera can reduce the amount of calculation [3], thereby greatly improving the speed and meeting the real-time requirements of the algorithm. But when detecting in a complex environment, interference such as background noise will be mixed with traffic signs. Hence, algorithms that only consider the color space cannot achieve good detection results. There are also many solutions. Zhou et al. used color threshold and shape analysis to segment traffic signs from images. Complementary data obtained from different sensors was utilized to fuse the prior location, color, laser reflectivity, and lidar data of traffic signs. The above operations improved the robustness of the algorithm [55]. Zhu et al. converted the image from the RGB model to the HSI model, and the red color was extracted from the H channel value. Then, the template LOG was used to extract the edge. Finally, the BP network was adopted to process the image [38]. However, converting RGB to HSI color space requires a certain amount of computation, which requires hardware processing to improve real-time performance.

2.2 Deep learning-based TSD

With the wide application of deep learning technology in various fields, it has demonstrated the powerful ability to learn feature representations from raw data. The representative network CNN is one of the most widely used network models for deep learning in computer vision [22]. Therefore, many CNN-based methods have been transformed to address the tas of TSD. Various object detection models were improved in [1] and applied to TSD. Sermanet et al. used a multi-scale CNN network for TSD and obtained an accuracy of 99.17% [36]. Belghaouti et al. proposed an automatic road sign recognition system based on the LeNet model, which achieved 99% accuracy in the German traffic dataset [2]. Song et al. advanced an efficient convolutional neural network (CNN) that can significantly reduce redundancy parameters, and increase the speed of the network [40]. Wang et al. proposed a new space-cover convolutional neural network (SC-CNN) for technological conundrum [46]. Zhou et al. proposed the Ice Environment Traffic Sign Recognition Benchmark (ITSRB) and Detection

Benchmark (ITSDB) annotated in the COCO2017 dataset format. They put forward an attention network-based approach for high-resolution traffic sign classification (PFANet) and performed ablation experiments on the designed parallel fused attention module [56]. Zhu et al. modified the OverFeat framework and proposed a single network that simultaneously detects and classifies landmarks [57]. Li developed a novel perceptual generative adversarial network to improve detection performance by generating super-resolution images of small traffic signs [19]. MR-CNN [25] adopts a multi-scale deconvolution structure that combines the features from deep and shallow layers. The fused feature maps reduce the number of region proposals to a certain extent and improve the efficiency of TSD. In [30], a feature aggregation structure is proposed to aggregate regional features of different scales, which improves the performance of small traffic signs. Zhang et al. proposed a cascaded R-CNN network for detecting small traffic sign instances and designed a data augmentation method to increase the number of difficult negative samples [53]. SADANet [26] combines a domain-adaptive network and a multi-scale prediction network to address the scale variation problem. The TSD method based on deep learning learns the features in a large amount of data, which has more advantages than the traditional method using artificially designed features. It is also not easily affected by external factors such as illumination and occlusion. Compared with the traditional detection method, the generalization ability is strong and accurate.

2.3 Multi-scale feature fusion

In the target detection task, the most important problem is how to extract target features more accurately [9]. Current neural networks like depth convolution structure used for feature extraction. With the deepening of network layer, the network reception field increases gradually, and the semantic expression ability also increases. But it also reduces the resolution of the images, and many details characteristics after multi layer network of convolution blur, such as smaller traffic signs. Shallow neural networks have smaller receptive fields and richer details, but weaker semantic information is extracted. In order to obtain accurate semantic information, traditional target detection models usually only use the feature graph output from the last layer of feature extraction network to classify and locate objects. However, the graph of the last feature corresponds to a large down-sampling rate, resulting in less effective information and reduced detection ability of small targets. Multi-scale feature fusion solves this problem well. FPN (Feature Pyramid Network) used the RPN to extract candidate regions on the feature pyramid [33]. By fusing deep features with shallow features, predictions were made at multiple scales of the feature pyramid. Thus, the semantics of shallow feature maps were enhanced, and the detection accuracy of small targets was improved. The study [44] proposed an efficient and accurate arbitrary-shaped text detector, termed Pixel Aggregation Network (PAN), which is equipped with a low computational-cost segmentation head and a learnable post-processing.

2.4 Research on model lightweight

Deep neural networks (DNNs) have recently achieved great success in many visual recognition tasks. However, existing neural networks require a lot of memory space and computational cost, making them difficult to deploy in devices with low memory resources. Solving these problems requires joint solutions from many disciplines, including but not limited to machine learning, system structure optimization and hardware design. Reference [16]

proposed to use different tensor decomposition schemes, which only lost 1% of the accuracy and achieved a 4.5x speedup. Since the translation-invariant property is ensured when exploring the features of the input image, the parameters of the CNN network are efficient, which is the key to successfully training the deep neural network and avoiding overfitting. Using a compact convolution kernel to replace a convolution kernel with a large number of parameters can directly reduce the amount of computation. SqueezeNet [47] adopts a 1 × 1 convolutional layer instead of a 3 × 3 convolutional layer, which reduces the number of parameters. The same approach is also adopted for MobileNets [13]. The work in [37] introduces a more advanced successor of the CNNS called 3-D CNNS, and the computing time (0.19 seconds per frame) of the proposed work shows that the proposal may be used in real-time applications. The work in [35] exploits the advantages of deep neural networks to solve the network compression problem. It proposes FitNets to train deep yet lightweight networks to compress large deep neural networks.

3 Proposed method

3.1 Overall architecture

The YOLOv5 [27, 42] network is the latest model in the YOLO series. The network model has high detection accuracy, fast inference speed, and the fastest detection speed can reach 140 frames per second. The weight file of the YOLOv5 network model is nearly 90% smaller than that of YOLOv4, which indicates that the YOLOv5 model is suitable for deployment on embedded devices for real-time target detection. However, YOLOv5 still has defects in the problem of small target detection, and cannot accurately identify smaller targets. To solve this problem, a multi-scale detection layer is further added in the latest YOLOv5 series named YOLOv5-P6 [42]. But what followed is a substantial increase in the amount of parameters and FLOPS. The YOLOv5 model has four architectures, named YOLOv5-s [42], YOLOv5-m [42], YOLOv5-l [42] and YOLOv5-x [42]. The main difference between them is the depth and number of feature extraction modules and convolution kernels at specific positions, and the size and model parameters of the four structural models are accordingly increased. This paper needs to identify many small targets and the intelligent driving system has high requirements on the real-time and lightweight performance of the recognition model. Hence, the accuracy, efficiency and scale of the recognition model are comprehensively considered in this paper, and the improved design is carried out based on the YOLOv5s architecture.

The overall framework is shown in Fig. 2, including the backbone and head. The head is composed of neck and detector. Features need to be extracted from the detected image and use the backbone for localization and classification so as to detect the location and class of landmarks. The backbone network consists of Focus, Ghost Convolution (GhostConv), GhostBottleneck with three convolutional layers (C3Ghost) and Spatial Pyramid Pooling (SPP). The first layer of the backbone network is the focus module, as shown in Fig. 3. This module performs a slicing operation on the 640 × 640 × 3 input image and divides it into 4 parts. The 4 parts complement each other and expand from 3 channels of the input image to 12 channels. Finally, convolution is performed on the generated new image. The Focus module reduces the cost of convolution. The method of reshaping tensor is used to downsampling and increase channels which reduces FLOPs and increases speed. The channel expansion algorithm adopted by this module is as follows:

Algorithm 1: The process of Channel Expansion.

Input: input data $X \in R^{c_1 \times h \times w}$

Output: output data $Y \in R^{c_2 \times h' \times w'}$

1. **for** $1, 2, \ldots, c_1$ do

take a value of every other pixel to form a new image

end

2. updating parameters: $c_2 = c_1 \times 4, h' = h/2, w' = w/2$

C3Ghost module refers to the structure of CSPNet [45] and combines GhostConv to perform convolution operation on images. The feature map of the base layer in one stage is divided into two parts, which realizes feature extraction by cheap means. At the same time, the probability of repetition is reduced in the process of information integration. Section 3.2 details the specific structure.

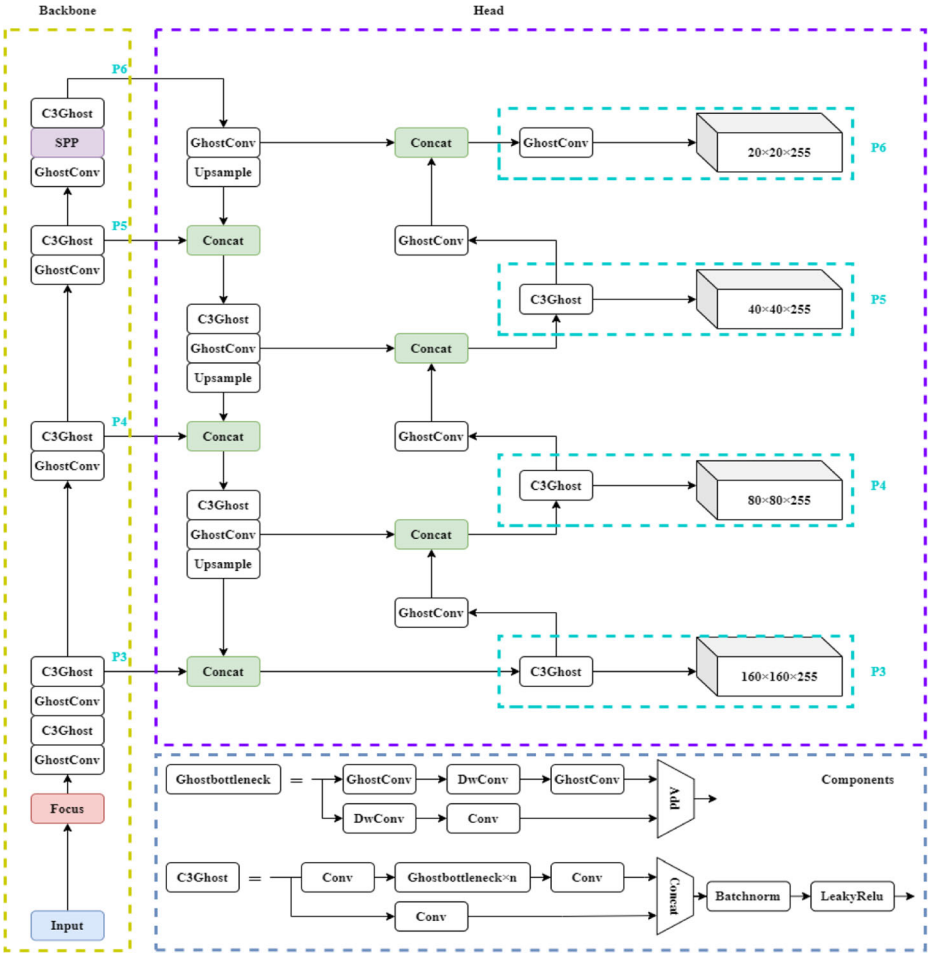


Fig. 2 The architecture of Ghost-YOLO

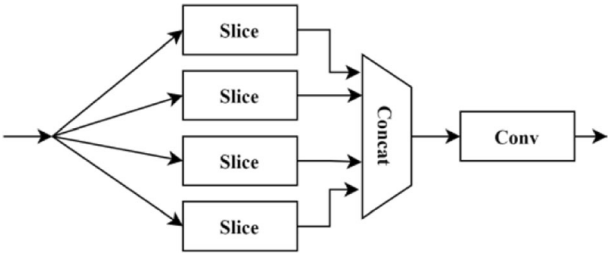


Fig. 3 Focus module

The last layer of the backbone network is the SPP module. After the combined three multi-scale max-pooling layers are used, the receptive field can be greatly improved with almost no speed loss while extracting features. It also effectively reduces the possible loss of image information when the images are directly stretched which ensures detection accuracy.

In the head part, the high-level feature information and the bottom-level feature information are transferred and fused by upsampling to realize a top-down information transfer structure. The Concat operation is performed on the bottom-level features and the high-level features so that the features with high resolution of the bottom-level can be easily transferred to the high-level. Thereby realizing the PANet [24] structure. Effectively utilize the complementary advantages of multi-scale features, and improve the accuracy of target recognition.

Based on YOLOv5s, the proposed model in this paper firstly replaces the convolutional layer and Bottleneck with the GhostConv module and the C3Ghost module to extract the features. Then add a detection layer of small target scale in the detector, which can more effectively identify small targets. Finally, the feature maps of each scale are input into the Detect module. In general, this method can validly enhance the recognition effect of small targets at high resolution. Moreover, it has less parameter quantity and calculation quantity, which achieves model compression under the premise of ensuring accurate detection. It is beneficial to deal with complex TSD scenes.

3.2 C3Ghost

CNNs have shown excellent performance in various computer vision tasks. The traditional CNNs usually require a large number of parameters and FLOPS to achieve satisfactory accuracy. Considering the extensive redundancy in the intermediate feature maps computed by mainstream CNNs, GhostNet [8] proposed an innovative convolution module, named Ghost module. It generates more feature maps to obtain the same effect as the original convolution through cheap linear operations. This new basic unit successfully achieves more feature maps with fewer parameters and computations, as shown in Fig. 4(a). Given input data $X \in R^{c \times h \times w}$ h and w are the height and width of the input data, while c is the number of channels of the input data. Any convolution operation used to generate n feature maps can be expressed as

$$Y = X \cdot \omega + b \tag{1}$$

where $Y \in R^{n \times h' \times w'}$ represents that the output is n feature map with height h' and width w', while $\omega \in R^{c \times k \times k \times n}$ represents that the convolution operation is performed by $c \times n$ convolution kernels of size $k \times k$, and b is the bias term. It is not difficult to find that the FLOPS required in

this convolution can be calculated as $n \cdot h' \cdot w' \cdot c \cdot k \cdot k$ This value usually reaches hundreds of thousands, because the number of convolution kernels n and the number of channels$c$are usually very large.

According to Formula 1, the dimensions of the input and output maps explicitly determine the number of parameters to be optimized (in w and b). Ghost module states that the feature maps generated by mainstream CNN operations contain a lot of redundancy, some of which are similar to each other. These redundant feature maps can be individually generated by using cheaper operations. As shown in Fig. 4(b), the feature extraction process of the Ghost module to generate m feature maps $Y' \in R^{n \times h' \times w'}$ can be expressed as:

$$Y' = X \cdot \omega' \tag{2}$$

where $\omega' \in R^{c \times k \times k \times m}$ represent the filters, $m \leq n$ and no bias term is required. Other hyperparameters such as convolution kernel size, stride and padding are consistent with ordinary convolution (Formula 1) to ensure the same size as the output feature map. A series of linear operations are adopted to generate repeating features according to the following formula:

$$Y_{ij} = \Phi_{i,j}\left(Y_i'\right), \forall i = 1, \ldots m, j = 1, \ldots s \tag{3}$$

where Y_i' represents the i-th feature map in Y', $\Phi_{i,j}$ is the j-th linear operation for each Y_i' to generate the j-th Ghost feature map Y_{ij}. The desired feature map can be obtained by directly

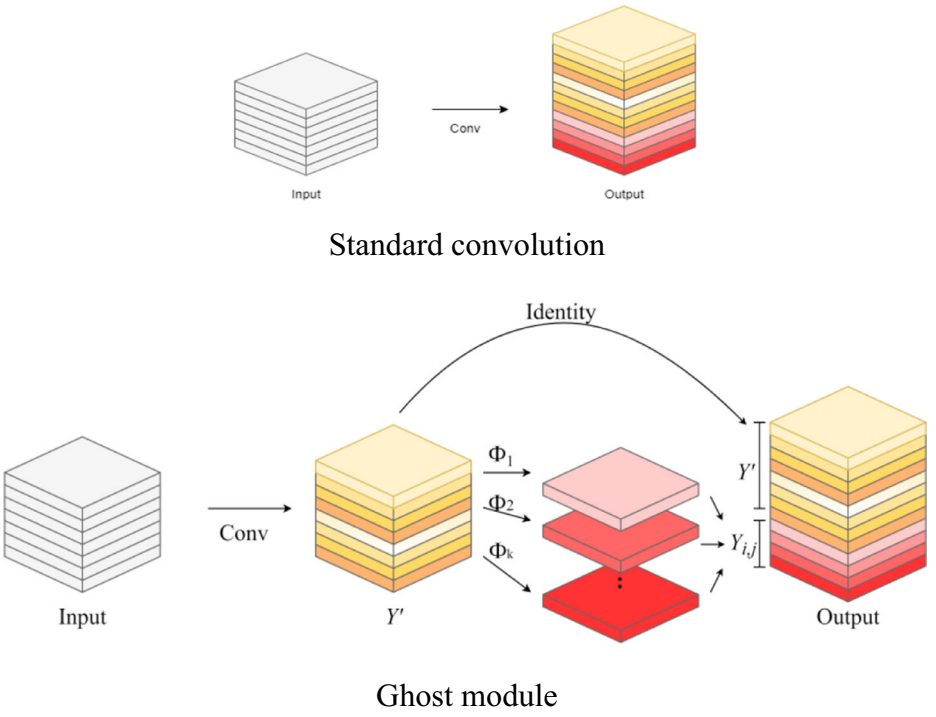


Fig. 4 Standard convolution and Ghost module

splicing the generated feature map with the feature map generated by the original convolution. After using Ghost module, set the linear convolution kernel size to $d \times d$, comparing the calculation amount of Ghost module and standard convolution can get the theoretical improvement degree.

$$C_s = \frac{c \cdot k \cdot k \cdot n \cdot h' \cdot w'}{\frac{n}{s} \cdot h' \cdot w' \cdot c \cdot k \cdot k + (s-1) \cdot h' \cdot w' \cdot d \cdot d} = \frac{c \cdot k \cdot k}{c \cdot k \cdot k \cdot \frac{1}{s} + d \cdot d \cdot \frac{s-1}{s}} \approx \frac{s \cdot c}{s + c - 1} \approx s \quad (4)$$

where $d \times d$ has a similar magnitude as that of k × k and s ≪ c, Thence, it can be quantitatively calculated that the calculation amount of the Ghost module is $1/s$ of standard convolution. The calculation of the parameters is similar, and it can also be simplified to s in the end. Theoretically, the superiority of the Ghost module can be quantitatively proved. So based on the Ghost module, the GhostBottleneck is designed. The specific structure is shown in Fig. 5(b).

Algorithm 2: Feature Extraction Based on C3Ghost.

Input: input data $X \in R^{c_1 \times h \times w}$, residual judgment value s, number of modules n
intermediate channels c'
Output: output data $Y \in R^{c_2 \times h \times w}$
1. Get X_1 in Conv1, $X \in R^{c' \times h \times w}, c' = c_2/2$
2. **if** $s = true$ do
Feature extraction via n GhostBottleneck modules
end
3. Get X_2 in Conv2 through the residual edge
4. $Y' = X_1\ contact\ X_2, Y' \in R^{c' \times h \times w}$
5. Send Y' to Conv3, get Y after BatchNorm2d and LeakyReLU, $Y \in R^{c_2 \times h \times w}$

Taking the advantages of Ghost module and GhostBottleneck, we introduced a lightweight feature extraction structure named C3Ghost. As shown in Fig. 5(c), it consists of three 1 × 1 convolution layers and n linearly stacked GhostBottleneck. c_1 and c_2 in Fig. 6 refer to the number of input and output feature map channels respectively, h and w have the same meaning as before. The first 1 × 1normal convolution is used to reduce the number of channels to 1/2 the number of output channels. Then features are extracted by linear stacked Ghostbottleneck and residual branches respectively. In this way, extracts the depth semantic information of the input image through two branches and the two sets of features are concatenated by contact module. Contact is a feature fusion operation, which splices two or more feature maps based on the number of channels and better utilizes the semantic information of feature maps of different scales to achieve better performance by increasing channels. Finally, the pieced signature information will pass through the BatchNorm module and use LeakyRelu as the activation function. The feature extraction Algorithm of C3Ghost is shown in Algorithm 2. In this process, the feature information of the original image is effectively preserved and the loss of features in the deep network is avoided. The C3Ghost module is applied to replace all BottleneckCSP modules in YOLOv5 to reduce the amount of computation and compress the model size. In theory, this method is completely feasible.

Algorithm 3: Training of Ghost-YOLO.

Input: Traffic sign dataset and annotation files

Output: Learned Ghost-YOLO model

Initialization: Epochs, Learning rate, Batch size, Input size, IoU threshold, Loss coefficient, Data enhancement coefficient, Label and anchor length, width ratio threshold.

1. Prepare the data, make the data set and divide the training set and validation set
2. Load the data configuration file and preset parameters
3. Load the model and perform feature extraction
   // training model
4. initialize the parameters W
5. **repeat**
6. find W with SGD
7. **if** epoch is not the last round
8. Calculate the mAP. If the calculated model has better performance. update and store the best model
   end
9. **until** getting the trained optimal model and the last trained model

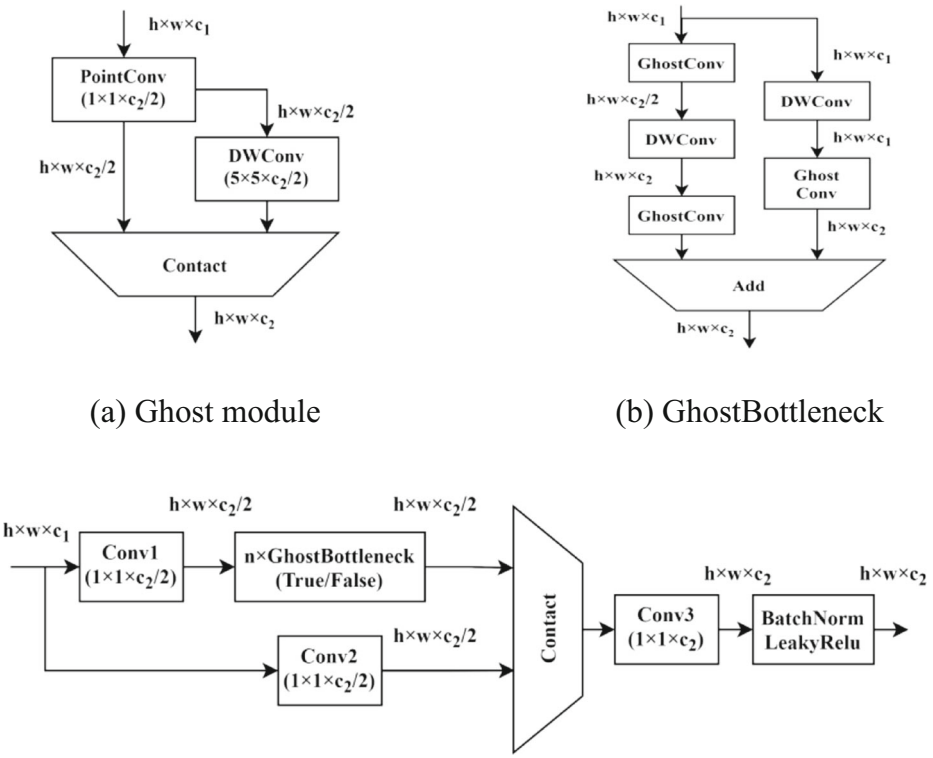


Fig. 5 The structure of Ghost module

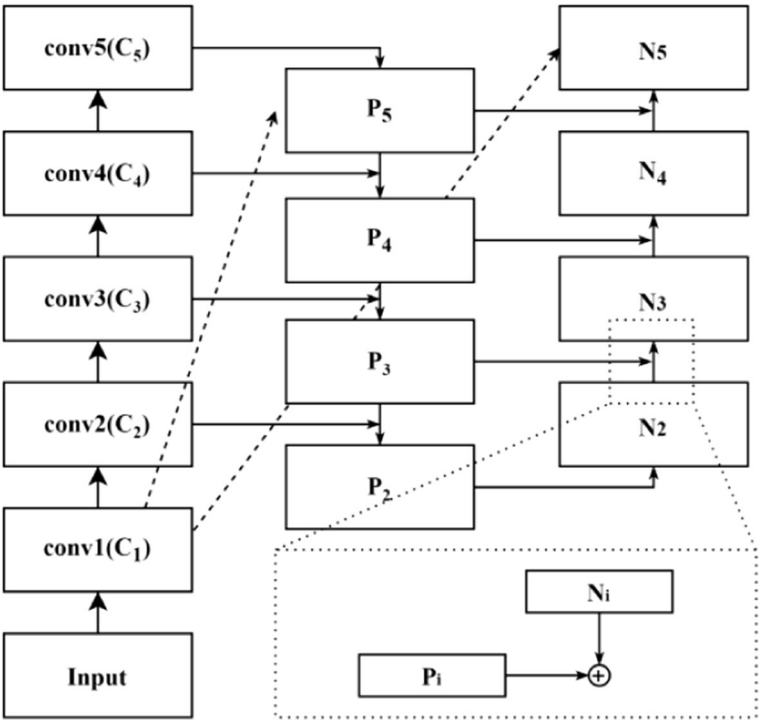


Fig. 6 The structure of multi-scale feature fusion module

3.3 Improvement of fusion feature layer

The fusion of features of different scales is a significant way to improve the recognition performance of the target detection network [21]. The purpose of feature fusion is to combine the features extracted from images into a feature with more discriminate ability. The current detection and segmentation networks mainly use convolutional networks to extract target features layer by layer. The low-level feature map has a higher resolution and contains more location and detail information. However, due to fewer convolutional layers, the lower-level feature map has less semantic information and contains more noise. High-level feature map has stronger semantic information, but due to the increased receptive field, the resolution of feature map is lower and the representation ability of geometric information is weakened. How to efficiently integrate the two is the key to improving performance.

Considering that there are many small targets in TSD, this study adds a multi-scale feature fusion detection module [29] based on the structure based on the model design in Section 3.3. As shown in Fig. 7, it consists of a top-down structure and a bottom-up structure. Firstly, feature extraction is performed on the input image, we can get [C_1, C_2, C_3, C_4, C_5] five groups of feature maps with different sizes. Through the up-sampling operation, the network obtains four groups of feature maps [P_5, P_4, P_3, P_2] from bottom-up paths, and obtains four groups of feature maps [N_2, N_3, N_4, N_5] from top-down paths. Unit addition is adopted in the fusion process, as shown in the dotted box. In addition, two shortcuts spanning multiple layers are included in the module to reduce information loss across layers. Finally, we can get feature maps of four scales.

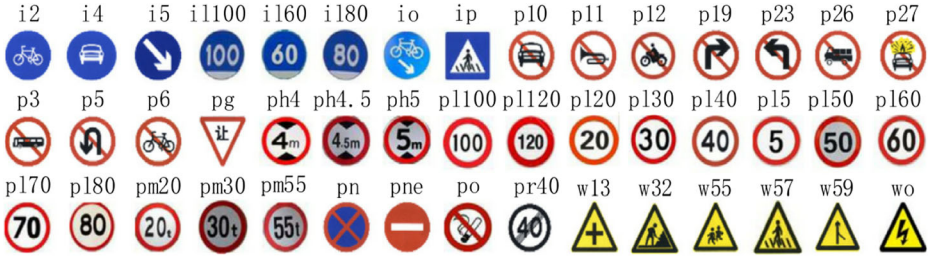


Fig. 7 Typical traffic sign categories in TT100K

Combined with the multi-scale feature fusion module to improve the network structure, the fusion of layers 4 and 15, 6 and 11, 10 and 21 in the original YOLOv5s architecture are changed to the fusion of layers 4 and 22, 6 and 18, 8 and 14, 16 and 28 in the network architecture designed in this paper. In order to improve the accuracy and make up for the loss of information caused by the low resolution of high-level features, the output features of the 20th and 25th layers of the improved network structure are fused.

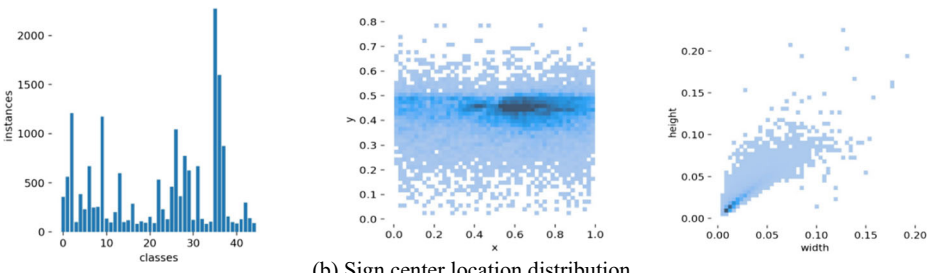
3.4 Training

Algorithm 3 describes the construction of the dataset and the complete training process. The design of hyperparameters will be given in Section 4.

4 Experiment

4.1 Data description

In this paper, we choose the TT100k [58] dataset as the experimental object. The TT100k dataset contains 9170 images, of which 6105 images are used as the training set and 4071 images are used as the test set. The size of the picture is 2048 × 2048 and includes the situation in different light and weather conditions. The size of the traffic sign is between 8 × 8 and 400 × 400, which is about 0.001%–4% of the whole picture. We ignore classes with less than 100 tt100k instances to ensure there is enough data for each type of traffic sign, leaving 45 classes for detection. Through the analysis of the data set, the visualization results are obtained, as shown in Fig. 8. The data set format is PASCAL VOC format, but YOLOv5 needs a txt tag



(a) Sign category distribution

(b) Sign center location distribution

(c) Sign size distribution

Fig. 8 Dataset analysis

file in YOLO format, and Ghost-YOLO also inherits this. The YOLO format is specifically (class_id, x, y, w, h), and they are all normalized results, so the original data set needs to be converted accordingly. The format conversion operation rules are as follows:

$$x = x_{center}/width \tag{5}$$

$$x = \mathrm{y}_{center}/height \tag{6}$$

$$w = (\mathrm{x}_{\max} - x_{\min})/width \tag{7}$$

$$h = (\mathrm{y}_{\max} - y_{\min})/height \tag{8}$$

Among them, $x_{\max}$ $y_{\max}$ $x_{\min}$ and $y_{\min}$ respectively refer to the coordinates of the upper left corner and the lower right corner of the position of the marked object relative to the upper left corner of the picture in the VOC marking format. These coordinates are given in the dataset.

4.2 Metrics and experiment setup

In the object detection task, total samples can be divided into three types. TP (true positive) denotes the targets which are correctly detected, FN (false negative) indicates the targets that have not been detected and FP (false positive) is used to denote the incorrect detections of targets. Three criteria are used to evaluate the performance in this study, including precision, recall and mAPs. Precision (P) [50] was used to evaluate the percentages of correct predictions in the results. Recall (R) [50] was used to evaluate how many positive samples were correctly detected. These two criteria are defined as follows:

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

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

Mean average precision (mAP) [50] is a commonly used metric to evaluate object detectors, as shown in formula 11. In both metrics, to be considered as a true positive, the intersection-over-union (IoU) overlap between the detection and the ground truth needs to exceed the defined minimal value. IoU was used to represent the overlap rate of predicted and real borders, which is the ratio of their intersection to union. We used two types of mAPs here. mAP_0.5 refers to the average AP of the classes when the IoU is set to 0.5, and mAP_0.5:0.95 refers to the average mAP at different IoU thresholds. The IoU value ranges from 0.5 to 0.95 with a step size of 0.05.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{11}$$

Table 1 Experimental environment

OS	Windows 10
CPU	Intel Core i5-10400F @2.90GHz
GPU	GTX 2070 SUPER
Memory	8 GB
CUDA version	11.1.0
Pytorch version	1.8.0

The experiment was run on a GPU server. Table 1 shows the detail of the experiment environment. We mainly use Python 3.8, Pytorch, OpenCV and other required libraries to implement our model.

The proposed Ghost-Yolo was using a backpropagation learning algorithm with CIoU (Complete-IoU) and BCE (Binary Cross Entropy) as the loss function and the stochastic gradient descent (SGD) algorithm as the optimizer. The model has about 30 hyperparameters for training settings, including training parameters and image processing parameters. The initialization values of key hyperparameters are given in Table 2. The training parameters include various coefficients and momentum and the image processing parameters include the coefficient of data enhancement. Learning rate is an important hyperparameter that cannot be ignored in model training, setting a proper learning rate can help model training. The learning rate of all experiments in this paper uses Warmup [11] to avoid model oscillation caused by a high initial learning rate during model training, cosine warmup is used to update the learning rate. The learning rate of the bias layer is decreased from 0.1 to the preset learning rate of 0.01, and the learning rate of other parameters is increased from 0 to 0.01, then attenuated according to the cos function value. All experiments are trained for 700 epochs.

4.3 Result analysis

To demonstrate the advantages of the proposed method in the task of TSD, we compare the proposed method with RentinaNet [20], Faster R-CNN [34], R-FCN [5], SSD [23], YOLOv3 [31], MSA_YOLOv3 [54], YOLOv4 [4], the original YOLOv5-s and our Ghost-YOLO. Among these models, Faster R-CNN represents a two-stage detector, while YOLOv3 is a representative one-stage detector. As can be seen from Table 3, RentinaNet achieves poor results in the two-stage detector, with an accuracy rate of only 69.83%. YOLOv3 is an efficient one-stage detector with results comparable to the faster Faster R-CNN. These demonstrate that simply applying a generic object detector to TSD does not achieve significant results. Ghost-YOLO proposed in this paper achieves competitive performance on the TT100k dataset. Using the Ghost-YOLO model, the accuracy rate is 93.48%, the recall rate is 89.65%, the mAP_0.5 is

Table 2 Settings of parameters

Parameter	Value
Momentum	0.937
Weight_decay	0.0005
warmup_epochs	3.0
warmup_momentum	0.8
Learning rate	0.01
IoU training threshold	0.20

Table 3 The recognition results of different methods on TT100K dataset

Method	Precision(%)	Recall(%)	mAP_0.5(%)	mAP_0.5:0.95(%)	FPS($f \cdot s^{-1}$)
RentinaNet	69.83	75.71	75.02	71.5	5.4
Faster R-CNN	74.12	87.95	85.84	77.95	3.5
R-FCN	76.69	89.16	87.19	85.28	2.8
SSD	70.41	76.05	75.11	71.6	7.4
YOLOv3	77.60	68.50	73.47	51.54	11.1
MSA_YOLOv3	–	–	86.30	–	23.8
YOLOv4	85.84	81.91	85.06	66.09	65.8
YOLOv5-s	88.30	85.52	89.45	70.27	48.3
Ghost-YOLO	93.48	89.65	92.71	73.31	56

92.71%, and the mAP_0.5:0.95 is 73.31%. Compared with the YOLOv5-s model, the accuracy rate increased by 5.2%, the recall rate increased by 4.13%, mAP_0.5 increased by 3.26%, and mAP_0.5:0.95 increased by 3.04%. For the two-stage detector, mAP_0.5 is 6.8% higher than Faster R-CNN and 5.52% higher than R-FCN. Compared with other legacy models, the Ghost-YOLO model is competitive in all detection metrics.

Moreover, some state-of-the-art methods were compared. As shown in Table 4, our method achieves similar or even better results than these latest methods. Finally, we show the detection results (mAP_0.5) of different algorithms in each category. We can see that methods such as Faster-RCNN perform better for large objects but less well for small objects. Our method achieves better performance in most categories, especially in small flag categories, such as 'wo', 'io', etc., Ghost-YOLO has a significant improvement, which also shows the improvement of multi-scale feature fusion for small target detection (Table 5).

In order to further confirm the efficiency of the modules and networks proposed in this article, as shown in Table 6, we present the mAP(0.5), speed and some other evaluation indicators of the improvement. We think that inference speeds faster than 30FPS can be considered real-time detection. It can be seen that the YOLO series has obvious advantages in speed, and the inference speed of our model can reach FPS 56. Yolov5-s can reach FPS 48.3, slightly lower than our model. Yolov3-tiny can reach FPS 60.4, which is the fastest. Ghost-YOLO also has advantages in network size, the calculation amount and parameter amount are compressed to 50.29% and 91.4% of the original, and the derivation process is improved by 6.25%. YOLOv3-tiny is a representative lightweight model, compared with it, the amount of computation and parameters are compressed to 65.64% and 75.8%. Although the inference speed is slightly slower, however, our model maintains high accuracy while performing real-time detection,

Table 4 Comparison with the state-of-the-art method on the TT100K dataset

Method	Year	mAP(%)	Precision(%)	Recall(%)	FPS($f \cdot s^{-1}$)
Zhu et al.	2016	81.56	–	–	–
Yang et al. [49]	2018	80.31	–	–	–
Wang et al. [43]	2020	–	93.80	92.25	–
Qing et al. [41]	2021	93.0	–	–	2.3
Liu et al. [28]	2021	93.0	–	–	20.6
Ours	–	92.71	93.48	89.65	56

Table 5 Comparison of mAP_0.5 for each class in TT100K dataset

Method/Category(%)	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	P19	P23	P26	P27
Faster R-CNN	68.0	62.0	71.0	52.0	63.0	76.0	51.0	48.0	54.0	69.0	73.0	67.0	94.0	67.0	67.0
Zhu et al.	76.7	88.2	93.5	98.5	91.0	93.5	82.0	89.4	85.7	90.0	88.5	67.8	90.4	87.2	86.1
SSD	70.1	79.3	85.3	77.1	86.4	78.7	72.3	71.6	64.5	57.1	67.7	73.0	80.4	70.7	76.2
YOLOv3	60.4	79.3	80.4	48.9	76.2	51.8	80.7	90.0	52.7	49.3	65.6	70.5	80.1	55.8	60.9
MSA_YOLOv3	86.5	90.0	94.0	90.2	96.0	92.8	87.0	86.4	76.9	75.3	80.9	79.0	85.8	83.4	82.5
YOLOv5-s	88.8	92.2	95.6	94.3	97.9	92.2	89.9	87.5	83.0	85.5	83.6	84.4	90.3	89.0	79.8
Ghost-YOLO	91.3	89.4	95.3	96.4	99.2	98.2	98.2	91.7	92.9	93.4	90.4	89.5	92.8	94.6	94.0
Method/Category(%)	p3	p5	p6	pg	ph 4	ph 4.5	ph 5	pl100	pl120	pl20	pl30	pl40	pl5	pl50	pl60
Faster R-CNN	62.0	92.0	66.0	44.0	94.0	75.0	63.0	81.0	91.0	88.0	73.0	85.0	89.0	76.0	86.0
Zhu et al.	85.1	91.9	87.0	82.0	87.6	88.0	85.4	97.5	99.0	92.9	92.0	92.4	88.7	90.4	93.0
SSD	66.5	74.9	63.9	84.2	62.1	51.2	78.6	85.1	84.2	45.4	66.6	65.7	60.5	58.3	64.0
YOLOv3	65.1	73.1	63.0	79.2	84.4	93.0	53.4	75.4	58.4	66.0	55.3	59.5	56.8	57.0	62.3
MSA_YOLOv3	82.5	84.2	74.9	92.9	77.2	85.4	67.6	88.5	82.9	78.3	77.8	75.5	81.8	75.3	76.0
YOLOv5-s	76.0	92.6	71.6	87.1	70.6	85.6	82.6	89.8	87.5	67.4	87.2	91.1	86.5	81.1	86.1
Ghost-YOLO	78.5	95.0	69.3	92.1	65.4	94.9	91.7	93.9	1.00	81.2	92.4	95.6	92.6	91.2	92.4
Method/Category(%)	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	wo
Faster R-CNN	67.0	76.0	93.0	77.0	57.0	86.0	21.0	33.0	10.0	36.0	30.0	70.0	38.0	53.0	16.0
Zhu et al.	94.0	94.5	89.5	85.7	73.5	91.5	93.0	74.5	85.6	65.0	79.0	78.4	86.3	78.3	48.2
SSD	70.6	70.5	69.6	51.3	71.2	71.7	86.4	51.8	87.9	46.1	57.1	64.6	74.0	58.8	39.7
YOLOv3	67.0	67.7	55.2	80.1	73.0	63.8	90.4	87.8	95.7	55.6	59.8	67.7	80.7	81.1	53.3
MSA_YOLOv3	81.0	77.1	73.9	76.1	77.0	86.4	96.5	74.5	88.4	73.8	84.6	81.0	88.4	81.4	54.9
YOLOv5-s	74.9	85.1	75.1	53.2	88.0	92.8	96.6	73.0	97.1	73.6	62.9	74.5	81.7	84.9	62.9
Ghost-YOLO	73.1	92.2	90.9	78.3	1.00	94.1	96.6	77.5	96.9	76.2	68.8	71.8	88.3	90.0	75.0

Table 6 Performance comparison of each model

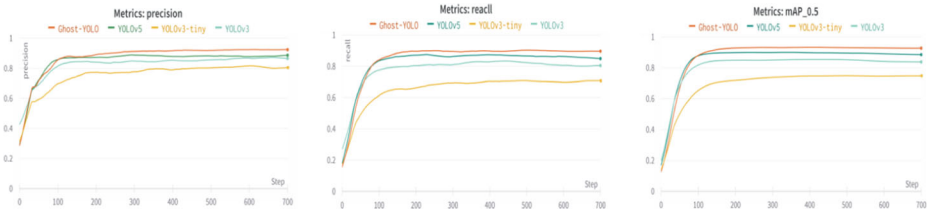
Method	mAP(0.5)	Params	Layers	GFLOPs	Speed (s)	FPS($f \cdot s^{-1}$)
Faster R-CNN	85.84	–	–	–	0.076	3.5
YOLOv3	73.47	61,760,674	333	155.7	0.034	11.1
YOLOv3-tiny	71.24	8,771,516	59	13.1	0.012	60.4
MSA_YOLOv3	86.3	–	–	–	0.042	23.8
YOLOv5-s	89.45	7,276,605	283	17.1	0.017	48.3
Ghost-YOLO	92.71	6,655,232	640	8.6	0.014	56

which proves that our method achieves a balance between accuracy and speed. Finally, we compare the improved method with the training process of YOLOv5-s and YOLOv3 which are representative models of the YOLO series and the comparable lightweight model YOLOv3-tiny. Figure 9 sketches these curves. Due to the learning of the model on the dataset, these values all increase rapidly. The changes stabilized at 100 epochs but were still gradually increasing, and all fluctuations were within acceptable limits. When the training ends at 700 epochs, the metrics of each model reach the maximum value. Ghost-YOLO also achieved the best performance.

In addition to the intuitive loss performance, this paper also draws the classification confusion matrix, as shown in Fig. 10. The confusion matrix is used to summarize the classification results and represent the accuracy evaluation matrix, the darker the color, the higher the recognition rate of the targets.

4.4 Ablation studies

In this section, we verify the impact of each component in Ghost-YOLO on the final performance, we conduct an ablation study on the TT100K dataset. The baseline is the original YOLOv5-s. As shown in Table 7, we compare the results by mAP(0.5) and FPS. Compared to the baseline, the model using only the C3Ghost module has a significant improvement in speed and a small loss of accuracy but is within acceptable limits. This is because Ghost Conv uses linear operation to replace the complex convolution operation, and the feature extraction effect has a certain fluctuation. Networks using the improved feature fusion module show significant improvements in mAP, but are slightly slower. This can be explained as follows: multi-scale feature fusion significantly improved the accuracy of small target recognition, but more complex feature fusion also reduced the inference speed.



Precision Recall mAP_0.5

Fig. 9 Performance comparisons of Ghost-YOLO, YOLOv5, YOLOv3 and YOLOv3-tiny

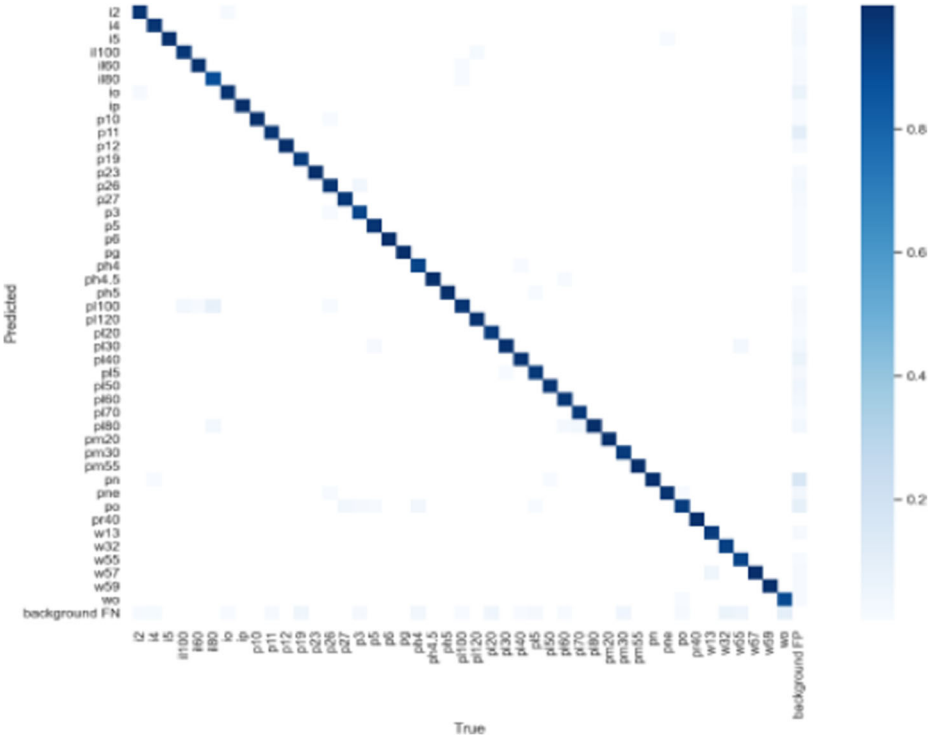


Fig. 10 Confusion matrix of Ghost-YOLO on test set

4.5 Visualization result

To directly verify the detection capability of the model, we visualize the results. Figure 11(a) gives the detection result of the YOLOv5-s and Ghost-YOLO. The original image is in the first column, the second and third columns respectively represent the visual detection results of the above two models. We can zoo min on the picture to see the more detailed detection results. It can be seen that both YOLOv5 and Ghost-YOLOv5 have good recall since they both detected the target. Ghost-YOLOv5 has a certain degree of accuracy improvement in various detection targets and is more accurate than YOLOv5 in recognizing farther and smaller traffic signs, which is useful for TSD tasks and driverless safety. Moreover, Fig. 11 also shows some detection results for traffic signs in complex environments such as occlusion and shadow and the results of small targets. It shows that our model excellently detects and recognizes traffic signs.

Table 7 Ablation studies of the proposed Ghost-YOLO, FF stands for the improved feature fusion structure

Model	mAP(%)	FPS($f \cdot s^{-1}$)
YOLOv5-s	89.45	48.3
YOLOv5-s+C3Ghost	86.87	88.0
YOLOv5-s+FF	93.2	47.3
YOLOv5-s+C3Ghost+FF(ours)	93.1	56.0

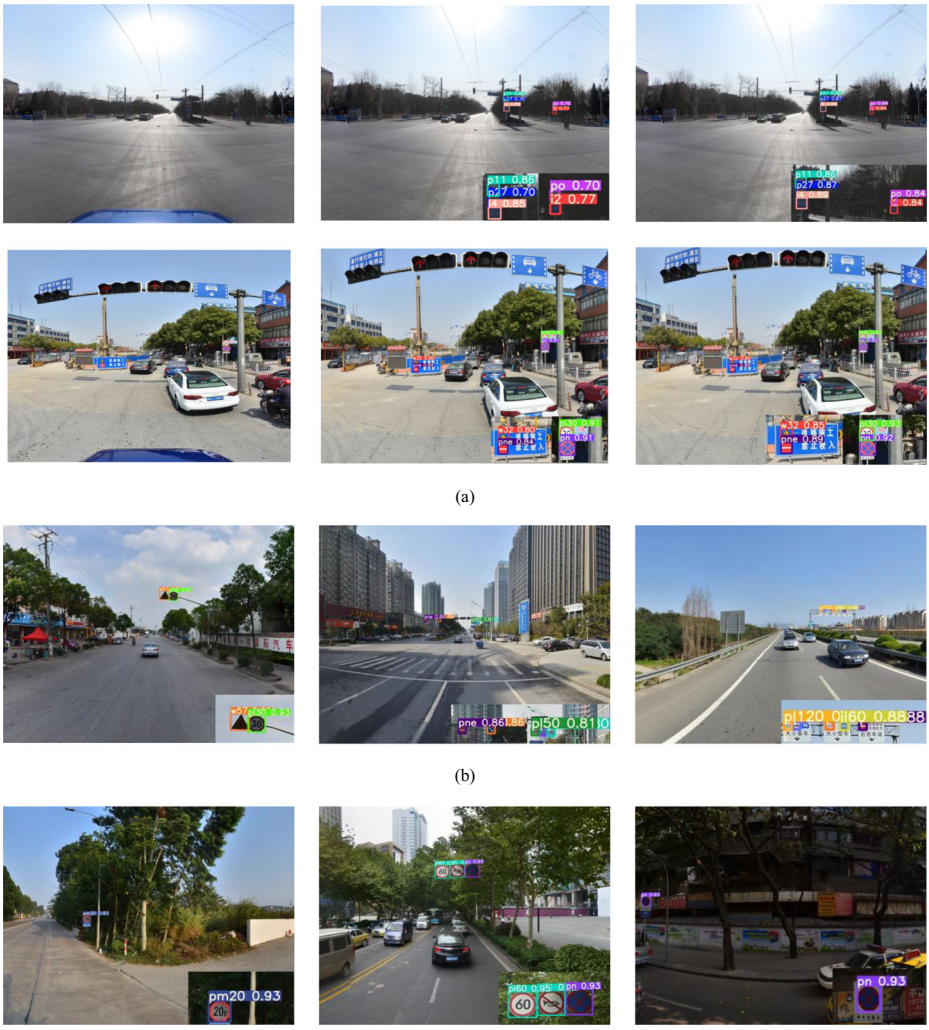


Fig. 11 Visualization results on the TT100K dataset: (**a**) detection results for YOLOv5-s and Ghost-YOLO; (**b**) detection results of the small target; (**c**) detection results in complex environments. Shadows, occlusions, cloudy, etc.

5 Conclusions

In this paper, based on the framework of YOLOv5, aiming at the difficulty of the traffic sign detection and the shortcomings of original YOLOv5, the lightweight network Ghost-YOLO is proposed. We design a new feature extraction module to reduce the account of redundant parameters and computation and speed up inference. At the same time, the multi-scale feature fusion structure is used to combine the high-level semantic information in the deep feature map with the shallow feature map to improve the feature representation of small targets and improve the accuracy of TSD. Experimental results on the TT100K dataset showed that the method achieves the balance of accuracy and lightness and has better robustness. In future

work, we plan to explore a more lightweight model and further address the efficiency of the model on the mobile terminal. At the same time, considering that traffic signs are usually stored in image format, image classification based on massive data has become one of the important topics [17], and the work in this paper can also be used as one of the future work ideas for image classification.

Acknowledgments This work has been supported by the National Science Foundation of China, No.31870532.

Data availability The data that support the findings of this study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Arcos-García Á, Alvarez-Garcia JA, Soria-Morillo LM (2018) Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing 316:332–344. https://doi.org/10.1016/j.neucom.2018.08.009
2. Belghaouti O, Handouzi W, Tabaa M (2020) Improved traffic sign recognition using deep ConvNet architecture. Procedia Comput Sci 177:468–473. https://doi.org/10.1016/j.procs.2020.10.064
3. Benallal M, Meunier J (2003) Real-time color segmentation of road signs. In CCECE 2003-Canadian conference on electrical and computer engineering. Toward a caring and humane technology (cat. No. 03CH37436) Vol. 3, pp 1823-1826. https://doi.org/10.1109/CCECE.2003.1226265
4. Bochkovskiy A, Wang CY, Liao HYM (2020) Yolov4: optimal speed and accuracy of object detection. https://doi.org/10.48550/arXiv.2004.10934
5. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29.
6. Ding Y, Ma Z, Wen S, Xie J et al (2021) AP-CNN: weakly supervised attention pyramid convolutional neural network for fine-grained visual classification. IEEE Trans Image Process 30:2826–2836. https://doi.org/10.1109/TIP.2021.3055617
7. Girshick R (2015) Fast RCNN, in 2015 IEEE International Conference on Computer Vision (ICCV) pp. 1440–1448
8. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C (2020) Ghostnet: more features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 1580-1589
9. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824
10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition pp. 770-778
11. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778
12. Howard A G, Zhu M, Chen B, Kalenichenko D, Wang W et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861
13. Howard AG, Zhu M, Chen B et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
14. Huang G, Liu Z, Van Der Maaten L, et al (2017) Densely connected convolutional networks. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708.
15. Iandola F N, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K (2016) SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. https://doi.org/10.48550/arXiv.1602.07360

16. Jaderberg M, Vedaldi A, Zisserman A (2014) Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866. https://doi.org/10.48550/arXiv.1405.3866
17. Kayhan N, Fekri-Ershad S (2021) Content based image retrieval based on weighted fusion of texture and color features derived from modified local binary patterns and local neighborhood difference patterns. Multimed Tools Appl 80(21):32763–32790
18. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25:1097–1105
19. Li J, Liang X, Wei Y et al (2017) Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1222-1230. https://doi.org/10.48550/arXiv.1706.05274
20. Lin T Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision, pp 2980-2988
21. Lin TY et al (2017) Feature pyramid networks for object detection, 2017 IEEE conference on computer vision and pattern recognition (CVPR) IEEE computer society
22. Liu XF, Xiong F (2020) A real-time traffic sign detection model based on improved YOLOv3. IOP Conf Ser: Mater Sci Eng 787:012034
23. Liu W, Anguelov D, Erhan D, Szegedy C et al (2016) SSD: single shot multibox detector. In European conference on computer vision. Springer, Cham. pp. 21-37
24. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8759-8768. 10.48550/
25. Liu Z, Du J, Tian F, Wen J (2019) MR-CNN: a multi-scale region-based convolutional neural network for small traffic sign recognition. IEEE Access 7:57120–57128. https://doi.org/10.1109/ACCESS.2019.2913882
26. Liu Z, Shen C, Qi M, Fan X (2020) SADANet: integrating scale-aware and domain adaptive for traffic sign detection. IEEE Access 8:77920–77933. https://doi.org/10.1109/ACCESS.2020.2989758
27. Liu Y, Lu B, Peng J, Zhang Z (2020) Research on the use of YOLOv5 object detection algorithm in mask wearing recognition. World Sci Res J 6(11):276–284. https://doi.org/10.6911/WSRJ.202011_6(11).0038
28. Liu Y, Peng J, Xue J-H et al (2021) TSingNet: scale-aware and context-rich feature learning for trafc sign detection and recognition in the wild. Neurocomputing 447:10–22. https://doi.org/10.1016/j.neucom.2021.03.049
29. Mei Y, et al (2020) Pyramid attention networks for image restoration
30. Ou Z, Xia F, Xiong B, Shi S et al (2019) FAMN: feature aggregation multipath network for small traffic sign detection. IEEE Access 7:178798–178810. https://doi.org/10.1109/ACCESS.2019.2959015
31. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement
32. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779-788
33. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. Adv Neural Inf Proces Syst 28
34. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28
35. Romero A, Ballas N, Kahou S E, Chassang A, Gatta C, Bengio Y (2014) Fitnets: hints for thin deep nets. https://doi.org/10.48550/arXiv.1412.6550
36. Sermanet P, LeCun, Y (2011) Traffic sign recognition with multi-scale convolutional networks. In: The 2011 International Joint Conference on Neural Networks pp 2809–2813. https://doi.org/10.1109/IJCNN.2011.6033589
37. Sharma S, Kumar K (2021) ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. Multimed Tools Appl 80(17):26319–26331
38. Shuang Z, Yi Z, Lu X (2006) Intelligent Approach for Triangle Traffic Sign Detection. J Image Graph 01(08):1127–1131
39. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
40. Song S, Que Z, Hou J, Du S, Song Y (2019) An efficient convolutional neural network for small traffic sign detection. J Syst Archit 97:269–277. https://doi.org/10.1016/j.sysarc.2019.01.012
41. Tang Q, Cao G, Jo KH (2021) Integrated feature pyramid network with feature aggregation for traffc sign detection. IEEE Access 9:117784–117794. https://doi.org/10.1109/access.2021.3106350
42. ultralytics. (n.d.) YOLOv5. Available online: https://github.com/ultralytics/YOLOv5
43. Wan J, Ding W, Zhu H, Xia M, Huang Z, Tian L, Zhu Y, Wang H (2020) An efcient small trafc sign detection method based on YOLOv3. J Signal Process Syst Signal Image Video Technol 93:899–911. https://doi.org/10.1007/s11265-020-01614-2

44. Wang W et al (2019) Efficient and accurate arbitrary-shaped text detection with pixel aggregation network, 2019 IEEE/CVF international conference on computer vision (ICCV) IEEE
45. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH (2020) 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, pp 390–391
46. Wang C, Ning X, Sun L et al (2022) Learning discriminative features by covering local geometric space for point cloud analysis. IEEE Trans Geosci Remote Sens 60:1–15. https://doi.org/10.1109/TGRS.2022.3170493
47. Wu B, Iandola F, Jin P H, Keutzer K (2017) Squeezedet: unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. In proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 129-137. https://doi.org/10.48550/arXiv.1612.01051
48. Xia Y, Xu W, Zhang L, Shi X, Mao K (2015) Integrating 3D structure into traffic scene understanding with RGB-D data. Neurocomputing 151:700–709. https://doi.org/10.1016/j.neucom.2014.05.091
49. Yang T, Long X, Sangaiah AK, Zheng Z, Tong C (2018) Deep detection network for real-life traffc sign in vehicular networks. Comput Netw 136:95–104. https://doi.org/10.1016/j.comnet.2018.02.026
50. Yuan X, Guo J, Hao X, Chen H (2015) Traffic sign detection via graph-based ranking and segmentation algorithms. IEEE Trans Syst Man Cybern Syst 45(12):1509–1521. https://doi.org/10.1109/TSMC.2015.2427771
51. Zhang Z, Huang K, Wang Y, Li M (2013) View independent object classification by exploring scene consistency information for traffic scene surveillance. Neurocomputing 99:250–260. https://doi.org/10.1016/j.neucom.2014.05.091
52. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 6848–6856. https://doi.org/10.48550/arXiv.1707.01083
53. Zhang J, Xie Z, Sun J, Zou X, Wang J (2020) A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. IEEE Access 8:29742–29754. https://doi.org/10.1109/ACCESS.2020.2972338
54. Zhang H, Qin L, Li J, Guo Y, Xu Z (2020) Real-time detection method for small traffific signs based on Yolov3. IEEE Access 8:64145–64156
55. Zhou L, Deng, Z (2014) LIDAR and vision-based real-time traffic sign detection and recognition algorithm for intelligent vehicle. In: 17th international IEEE conference on intelligent transportation systems (ITSC), pp. 578–583. https://doi.org/10.1109/ITSC.2014.6957752
56. Zhou K, Zhan Y, Fu D (2021) Learning region-based attention network for traffic sign recognition. Sensors 21(3):686. https://doi.org/10.3390/s21030686
57. Zhu Y, Zhang C, Zhou D, Wang X, Bai X et al (2016) Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing 214:758–766. https://doi.org/10.1016/j.neucom.2016.07.009
58. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2110-2118

Shuo Zhang received the B.Eng. degree in communication engineering from Jishou University, JiShou, China, in 2019. He is currently pursuing the M.E. with the School of Computer and Information Engineering, Central South University of Forestry and Technology. His research interests include deep learning, neural networks, and object detection.

Shengbing Che received his Master's degree in Computer Application Technology from the School of Computer and Communication Engineering, Changsha University of Technology in 2005.Currently he is a professor in the School of Computer and Information Engineering, Central South University of Forestry and Technology. His main research interests are network information security, artificial intelligence, and digital image processing.

Zhen Liu received the B.Eng. degree in communication engineering from Hunan University of Humanities, Science and Technology, LouDi, China, in 2020. She is currently pursuing the M.E. with the School of Computer and Information Engineering, Central South University of Forestry and Technology. Her research interests include deep learning, neural networks, and object detection.

Xu Zhang received the B.Eng. degree in Wangjiang College of Ahui Normal University in 2021. He is currently pursuing the M.E. with School of Computer and Information Engineering, Central South University of Forestry and Technology. His main research interests are natural language processing and Chinese named entity recognition.