ITC 4/52

Information Technology and Control
Vol. 52 / No. 4 / 2023
pp. 966-983
DOI 10.5755/j01.itc.52.4.34039

Traffic Sign Detection Algorithm Based on Improved Yolox

Received 2023/05/05 | Accepted after revision 2023/09/30

HOW TO CITE: Xu, T., Ren, L., Shi, T.-W., Gao, Y., Ding, J.-B., Jin, R.-C. (2023). Traffic Sign Detection Algorithm Based on Improved Yolox. *Information Technology and Control*, 52(4), 966-983. https://doi.org/10.5755/j01.itc.52.4.34039

Traffic Sign Detection Algorithm Based on Improved Yolox

Teng Xu
School of Computer Science and Software Engineering, University of Science and Technology Liaoning,
114051 Anshan, Liaoning, China; e-mail: 1220175209@qq.com

Ling Ren
School of Innovation and entrepreneurship, University of Science and Technology Liaoning,
114051 Anshan, Liao-ning, China; e-mail:176878392@qq.com

Tian-Wei Shi
School of Computer Science and Software Engineering, University of Science and Technology Liaoning,
114051 Anshan, Liaoning, China; e-mail: tianweiabbcc@163.com

Yuan Gao, Jian-Bang Ding, Rong-Chen Jin
School of Computer Science and Software Engineering, University of Science and Technology Liaoning,
114051 Anshan, Liaoning, China

Corresponding author: 176878392@qq.com

This paper proposes a novel PVF-YOLO model to extract the multi-scale traffic sign features more effectively during car driving. Firstly, the original convolution module is replaced with the Omni-Dimensional convolution (ODconv) and the feature information obtained from the shallow feature layer is incorporated into the network. Secondly, this paper proposes a parallel structure block module for capturing multi-scale features. This module uses the Large Kernel Attention (LKA) and Visual Multilayer Perceptron (Visual MLP) to capture the information generated by the network model. It enhances the representation ability of feature maps. Next, in the process of training, the gradient concentration algorithm is used to optimize the initial Stochastic Gradient Descent (SGD). Under the condition of real-time detection, it improves the detection accuracy. Finally, to improve the robustness of the model, this paper conducts extensive experiments. Tsinghua-Tencent 100K (TT100K), Changsha University of Science and Technology CCTSDB (CSUST Chinese Traffic Sign Detection Benchmark) are used as the training data set. It verifies that the PVF-YOLO method proposed in this paper enhances the detection ability of traffic signs of different scales, and the detection speed and accuracy are better than the original model.

KEYWORDS: Traffic sign detection, Feature fusion, PVF-YOLO algorithm, Traffic sign Data enhancement, Gradient optimization.

1. Introduction

In recent years, intelligent driving has garnered considerable attention, and the automatic detection of traffic signs has emerged as prominent research focus both domestically and internationally. Traffic sign detection is a critical task in intelligent driving systems, as it involves accurate identification of the spatial position and category of traffic signs on the road ahead, enabling appropriate responses from drivers or central control platforms [9]. Although intelligent driving technology has matured, the occurrence of traffic accidents due to incorrect traffic sign recognition continues to pose a safety risk for road users.

In the past, traffic sign detection methods relied on template matching, which entailed utilizing pre-detected images to find matching targets in templates and subsequently annotating their coordinate positions [19]. These template-based recognition methods exhibit limited robustness and frequently falter in cases of image distortion or defects.

With the evolution of machine learning, algorithms that integrate feature extraction and machine learning have gained prominence. Although conventional two-stage detectors like Faster R-CNN [15] and YOLO [14] have demonstrated satisfactory results within experimental environments, their practical applications still pose challenges. The key issues in traffic sign detection revolve around ensuring both detection speed and accuracy. Conventional gradient optimizers encounter difficulties in enhancing training accuracy. In real-world scenarios, challenging weather conditions and rapid changes in traffic sign features can have a negative impact on the performance of the detector. The pursuit of real-time object detection, without compromising accuracy, has emerged as a significant hurdle in this domain. In light of these challenges, this paper proposes the PVF-YOLO object detection network, presenting the benefits of superior detection accuracy and fast recognition. Specifically, the contributions outlined in this paper are as follows:

1. Incorporating the shallow feature layer into the neck of the network and replacing the original convolution by Omni-Dimensional convolution (ODconv) to extract information from the shallow feature layer of the network.
2. Introducing a novel parallel visual feature module designed to extract deep abstract feature information. This module adeptly captures both global and local features of the target object, seamlessly integrating into the architecture of the network.
3. Drawing inspiration from gradient centralization, experimental updates have been made to the gradient optimizer of the YOLOX network, leading to a significant enhancement in detection accuracy.

The remainder of this paper is organized as follows: Section 2 delves into the discussion of related work, Section 3 elucidates the methodology employed, Section 4 outlines the conducted experiments, and finally, Section 5 presents the conclusions derived from the study.

2. Related Works

2.1. Deep Learning Based Traffic Sign Recognition Algorithm

Deep learning has powerful feature learning capabilities. Deep Convolutional Neural Networks (CNNs) do not require manual feature design; they perform supervised learning on input model images, completing feature extraction and classification with higher recognition rates than traditional algorithms such as AdaBoost and SVM.

Traffic sign detection based on deep learning typically uses CNNs to extract features from images. Object detection is then executed based on the obtained image features. The successful application of convolutional neural networks demonstrates their extraordinary potential [12]. Ciresan et al. [4] achieved a remarkable 99.46% accuracy on the GTSRB dataset using multi-column deep neural networks. However, the computational cost of this model is high, as it requires a large number of multiplication operations on hardware, and the efficiency of the activation functions used is also low. To alleviate the computational burden, Aghdam et al. [1] opted for the Rectified Linear Unit (ReLU) activation function and divided the two intermediate convolutional pooling layers into two groups, thereby halving the number of parameters in these layers. This streamlined model eliminates redundant neural network parameters, resulting in a recognition rate of 99.51%. The introduction of the YOLO series of single-stage object detection models has greatly advanced traffic sign recognition. Zhang et al. [25] proposed the CCTSDB dataset and improved the number of convolutional layers in the YOLOv2 network to better adapt to traffic signs. Zheng et al. [26] integrated YOLOv4-tiny into a low-cost embedded system. Efficient real-time target detection was achieved by thermal imaging capture via a camera. Wang et al. [20] proposed an improved Feature Pyramid model based on YOLOv5, which improves the detection accuracy of the model by utilizing an adaptive attention module and feature enhancement module while maintaining real-time detection. Recent research has shown that for single-stage object detection networks, the feature fusion stage is crucial to the results. Tan et al. [18] introduced learnable weights to adjust the features of different inputs, while employing bidirectional fusion to improve detection accuracy and efficiency. Zhu et al. [28] used adaptive attention modules and feature enhancement modules to reduce information loss during the feature map generation process. A multi-scale transformer with dual-channel representation was designed by Zheng et al. [27]. By introducing multiscale analysis into the model, the form the tighter decision boundary.

2.2. Traffic Sign Dataset

Innovation in autonomous driving algorithms relies on reliable traffic sign datasets. Therefore, the comprehensiveness of datasets is crucial as a primary factor affecting safe driving. To promote research in traffic sign detection, research institutions worldwide have compiled traffic sign databases, which serve as fundamental support for evaluating and comparing the effectiveness of various traffic sign recognition algorithms. In recent years, large-scale traffic sign databases have been created, as shown in Table 1, providing a foundation for researchers to develop new algorithms. These publicly available databases contain various traffic sign samples captured by cameras under various occlusion conditions, with sign shapes reflecting the diversity of real-world scenarios.

Table 1
Published traffic sign datasets

Dataset	Applications	Number	Categories	Country
GTSRB	Identification	51839	43	Germany
GTSDB	Detection	900	43	Germany
BTSD	Detection	9006	62	Belgium
LISA	Detection+ Identification	6610	47	America
CCTSDB	Detection+ Identification	17856	3	China
TT100K	Detection+ Identification	10000	45	China
GLARE	Detection	2198	47	America

Although most datasets contain traffic signs of different shapes and sizes, they overlook traffic signs collected under harsh weather conditions. Guo et al. [7] argue that adverse weather can have detrimental effects on traffic sign recognition. Therefore, creating a dataset that includes traffic signs under harsh weather conditions is particularly important.

GLARE is the first dataset containing traffic signs under bright lighting conditions. When testing models using datasets with strong light backgrounds, detection accuracy is significantly lower than when using regular datasets. Thus, it is especially important to use datasets containing traffic signs under adverse weather conditions. Harsh environments, such as heavy rain, snow, fog, or extreme lighting conditions, are part of real-world driving scenarios. Ignoring these environments in data collection leads to an incomplete representation of the challenges that autonomous vehicles and traffic management systems might face in practical situations.

3. Methodology

3.1. Architecture

As the latest model of MEGVII technology, YOLOX [7] has high accuracy, fast detection speed and easy to deploy. YOLOX has different versions, such as YOLOX-S, YOLOX-M, and YOLOX-L. This paper uses the YOLOX-S model as the improved baseline. Because of its minimal memory footprint, it has the potential for actual deployment in vehicles. YOLOX uses Cross Stage Partial Network (CSPNet) as the backbone network as shown in Figure 1. CSP is used to segment input data and connect it through multiple convolution layers to form input for subsequent layers. CSPNet reduces the calculation cost and parameter number of the model, and enhances the accuracy and robustness.

In the first stage, a locally dense block is divided into a partially dense block and a partially transitional layer. The base layer feature map for a phase is divided into two parts $x_0 = [x'_0, x''_0]$. x''_0 is directly connected to the end of the stage, and x'_0 will go through a dense block. In Figure 1, grey blocks are used to represent x''_0 and blue blocks to represent x'_0. The division of the inputs to the network makes it possible to reduce the computation of repeated gradients when updating the gradient weights.

In order to use the characteristic values passed by the backbone, the network structure must include a neck module. Its function is to add more contextual information to the feature graph to help the network better understand the image content. The neck module of YOLOX-S uses Path Aggregation Feature Pyramid Network (PAF-PN) structure to extract features from multiple feature maps at different scales and aggregate them to improve the network's understanding of image content. This method integrates high and low level semantic information, enhances feature representation ability, provides more effective information for network output and improves network performance. In addition, the decoupling detection head is used to solve the classification and regression conflict problems in the process of object detection. The model will also compare the overlapping areas between the prior box and the real box to determine whether the sample in the box is a positive/negative sample. YOLOX introduces an adaptive anchor generation mechanism that dynamically adjusts anchor scales to better match the distribution of object sizes in the dataset. This adaptive approach optimizes anchor selection and enhances detection performance, particularly in scenarios with skewed object size distributions.

The overall architecture of the PVF-YOLO proposed is shown in Figure 2. The red positions are the innovations proposed in this paper. ODConv is added to the Backbone and PVF block is added to the Neck of the model.

Figure 1

Cross Stage Partial Network

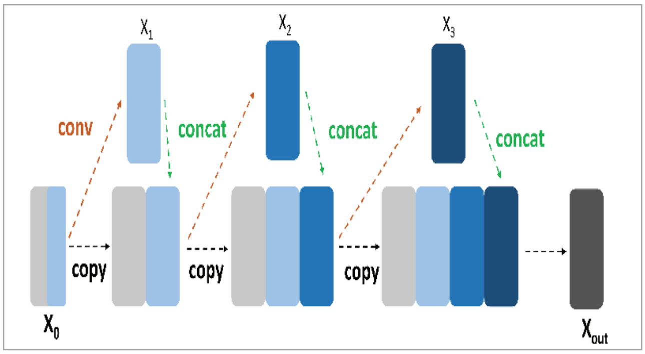
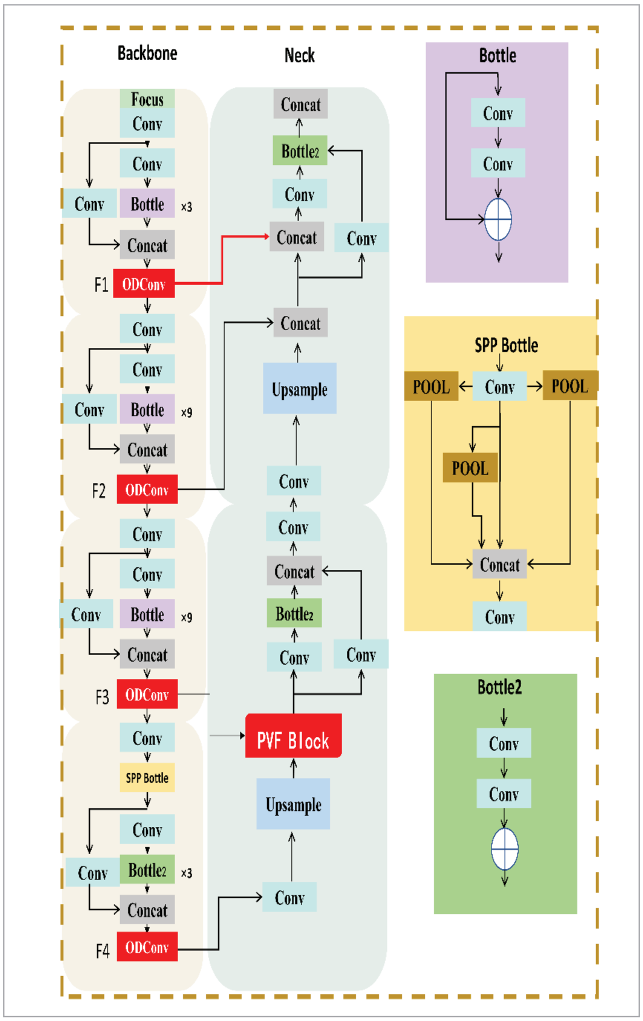


Figure 2

Overall architecture of the PVF-YOLO model. The Batch Normalization and silu layers have been omitted for intuitively illustration of the improvements. The Bottle structure is shown on the right



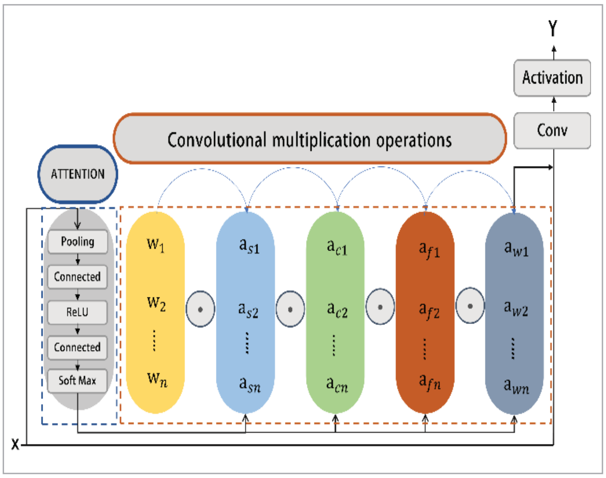
3.2. Innovative Methods

3.2.1. Shallow Feature Extraction

Multi-scale target detection in autonomous driving often requires a comprehensive image feature map with rich semantic information. To enable the model to grasp complete feature information, this paper employs ODconv, as depicted in Figure 3, to replace the original ordinary convolution. ODconv utilizes a multi-dimensional attention mechanism to output spatial dimensions and captures the global feature relation of the convolution kernel more comprehensively. Through the aggregation of multiple convolution kernels, the feature learning ability of the model is enhanced without increasing the number of layers. Experimental results demonstrate that ODconv can enhance the feature learning ability of the model [12].

Figure 3

ODconv Schematic. The comprehensiveness of kernel space information is ensured by using the four-dimensional parameters of the convolution kernel for computation



Define w_i as the attention scalar, a_{si} as the attention scalar of the convolution, a_{ci} as the attention scalar for different channels of the convolution filter, a_{fi} as the attention scalar for convolution filters, and a_{wi} as the attention scalar for the entire convolution kernel where n represents the range of values. The output is defined in Equation (1).

$$y = \{(w_1 \odot a_{s1} \odot a_{c1} \odot a_{f1} \odot a_{w1}) + \cdots + (w_n \odot a_{sn} \odot a_{cn} \odot a_{fn} \odot a_{wn})\} * x \quad (1)$$

As the attention scalar of ODconv varies with the input, it is represented as a weighted sum of multiple features for different kernel filters. Therefore, the weight value is a nonlinear function. In comparison to the linear weights of classical convolution, nonlinear scalars allow for a more comprehensive optimization of the network output [2]. When compared to static convolution linear function, ODconv evidently possesses a stronger feature learning capability. It can increase the complexity of the model without increasing the depth or width of the network, thereby enhancing the accuracy of Convolutional Neural Networks (CNNs) while maintaining efficient inference. The ODconv and conventional convolution are, respectively, defined as shown in Equations (2)-(3).

$$ODconv_{out} = \beta\big((K_1 \odot \alpha_{W1} + \cdots + K_n \odot \alpha_{Wn}) \cdot x\big) \quad (2)$$

$$conv_{out} = \beta(K_1 \cdot x) \odot \alpha_{W1} + \cdots + \beta(K_n \cdot x) \odot \alpha_{Wn}, \quad (3)$$

where β is the activation function, n is the number of experts, and $K_i (i \leq n)$ is the convolution kernel. Comparing (2) and (3) reveals that Odconv entails the same computational complexity as the conventional static convolution. However, since Odconv only computes complex convolution once, it is notably faster.

After optimizing the network output with Odconv, the propagation path of eigenvalues of the network is further updated to extract the previous shallow features. Through slicing, one-step convolution, and one-step CSP, the YOLOX-S network obtains the shallow feature layer F1. This layer encompasses more original and fuzzy feature information, possesses a smaller sensitivity field, and contains more global features and fine-grained information. However, the initial YOLOX-S network did not integrate the output of F1 with the neck network, resulting in the loss of crucial information. During the downsampling process, the receptive field for small objects continually expands, leading to subpar performance in capturing detail features and detecting small objects. Therefore, this paper establishes a connection between the shallow feature layer F1 and the neck network, thus enabling fusion of the shallow features captured by the model. Denote the generated feature mapping as M^k. Its feature map at position (©, j) is represented as ©. This classification is labeled as C. Following further pooling of the resultant feature maps, a linear transformation is applied based on the feature weight of each class, resulting in the feature map denoted as W_C^k. The weight A^C of this classification is derived as Equation (4):

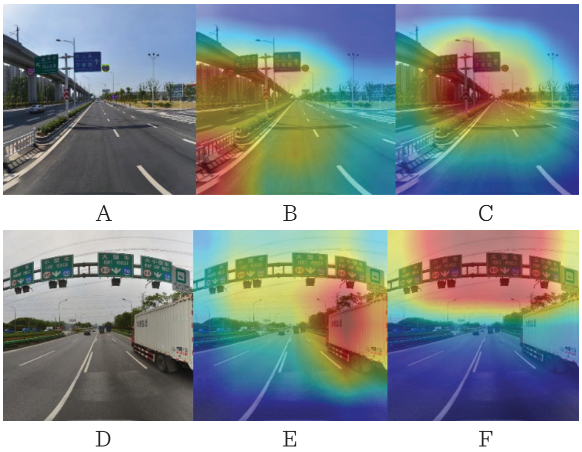
$$A^C = \sum_k W_c^k \frac{1}{z} \left(\sum_i \sum_j ©_{i,j}^k \right). \tag{4}$$

Regarding the classification weight A^C, when the small-size target feature information conveyed by the shallow feature layer is disregarded, © diminishes, leading to a corresponding reduction in the classification weight. Consequently, the model learns fewer features for that specific class, thus impacting the detection of small-size targets. To provide an intuitive explanation of this process, this paper employs Gradient-weighted Class Activation Mapping (Grad-CAM) to study the focus area of the model as depicted in Figure 4. Grad-CAM [17] is a gradient-based visualization technique for understanding the region of interest (ROI) within deep neural networks, particularly in image classification tasks. Grad-CAM assigns weights to detection classifications and generates heatmaps based on these weights. In these heat maps, the darker the color of the region, the more pronounced the detection capability of the model for that region.

In this paper, the shallow and deep feature layers of the path convergence network (PANet) are fused. This enables the model to transfer strong location information and edge feature details from the shallow layer to the deep semantic layer, enhancing gradient information and thereby improving target detection performance.

Figure 4

Comparison of feature extraction effects. (A)(D) is the original image, (B)(E) shows the effect of the original YOLOX network model, and (C)(F) shows the feature extraction effect after the fusion of the shallow feature layer F1 and the neck network. On the basis of retaining shallow features, the improved network updates the acquired feature map through weight adjustment during backpropagation, thereby enhancing the detection of distant traffic signs



3.2.2. Parallel Visual Feature Block

As the network deepens, the feature map becomes smaller and contains more abstract and high-level semantic information. However, the upsampling of feature information by the maximum pooling layer might lead to the loss of valuable spatial and semantic information within the feature map. Therefore, it is crucial to retain both spatial and semantic information during the design of deep neural networks. During vehicle operation, the object scale of traffic sign detection varies, which can significantly impact the performance of deep learning models. Rapid changes in traffic sign features can cause the loss of semantic information, disconnecting the relationship between feature information and image content. The complexity of convolutional neural networks is influenced by various factors like layer count, filters, and input. While increasing network complexity can improve feature extraction and training performance, it also raises the risks of overfitting and higher computational costs. Hence, striking the right balance between complexity, accuracy, and efficiency is pivotal for designing effective convolutional neural networks.

To capture abstract information within the deep layers of the model, this paper introduces the PVF module into the deep layer of the neck network. The abstract feature maps generated by F3 and F4 layers are fed into the PVF module. The deep features produced by PVF are then combined with the shallow features from ODconv, and the remaining network layers are adjusted using this merged multi-level feature set. The PVF module can capture both global and local features, seamlessly merging them into a cohesive feature representation. Initially, multi-scale features are extracted from the modified Backbone feature layer. Then, the complementary large kernel attention module and the visual Multilayer Perceptron (MLP) module are applied to capture global and local features, respectively, as illustrated in Figure 5.

The large core attention module calculates the channel attention graph for the entire feature graph, emphasizing informative features while suppressing noise. The Visual MLP module is a new Transformer architecture that incorporates an enhanced depth separable convolutional layer and a channel MLP layer. Finally, the feature graphs obtained from both modules are concatenated along the channel dimension to create a unified feature graph, which serves as the output of the PVF module. This process can be expressed as in Equation (5):

$$out=\text{torch.cat}(\text{LKA}_{out},\text{Visual Mlp}_{out}), \quad (5)$$

Where torch.cat is used to concatenate multiple tensors along a specified dimension. The dim parameter is omitted, with the default value of 0, which concatenates two vectors along the first dimension. Given the different tensor dimensions of the outputs of the PVF and Vision MLP modes, concatenation is required.

The resultant series feature map encompasses both global features and local details. Moreover, since each module only processes a subset of features, the use of parallel feature modules effectively reduces the computational cost and memory consumption of the PVF model.

– **Large Kernel Attention (LKA) module**

The self-attention mechanism constitutes an adaptive selection process. A pivotal aspect of this mechanism involves generating a feature map that highlights the significance of various parts within the input. While the conventional self-attention mechanism can establish extensive dependencies, it is accompanied by high computational complexity due to the necessity of computing attention weights for every feature position. Additionally, the attention weights are obtained through the SoftMax function, which can result in slower training. To solve the problems and limitations of the traditional self-attention mechanism, this paper adopts the LKA module in Figure 5 to capture the long-distance relationship in the image by decomposing the large convolution kernel transmitted by the model. This module avoids the softmax operation and reduces the amount of computation. The LKA module shares a similar design concept with Self-Attention, as depicted in Figure 6(b). The receptive field resulting from the combination of three different convolutions resembles that of a large convolutional kernel.

The feature map by the LKA module as in Equation (6):

Figure 5

General view of the PVF module

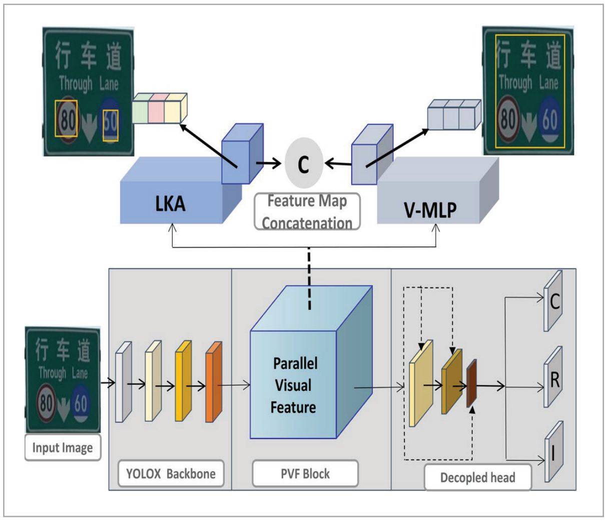
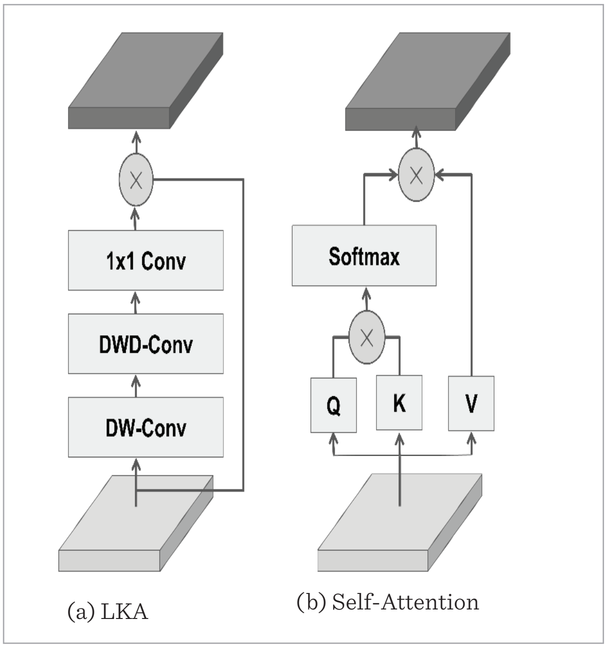


Figure 6

Standard LKA module and self-attentive module



(a) LKA (b) Self-Attention

$$\text{LKA}_{l} = \{[DW - Conv(P)]DWD - Conv\}1 \times 1Conv, \quad (6)$$

where A_1 is the feature map, p∈R is the input feature. For self-attention, Q,K,V are defined to represent query vectors, key vectors, and value vectors, where feature map $Attention_l$ can be represented as in Equations (7)-(8):

$$\mathrm{Attention}_l=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d(K)}}\right)V \qquad (7)$$

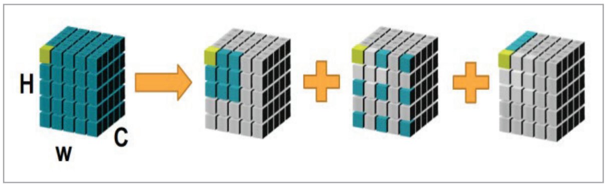
$$\mathrm{softmax}(Z_i)=\frac{\exp(Z_i)}{\sum_j \exp(Z_j)}. \qquad (8)$$

By avoiding complex operations, the LKA module can decrease memory usage and lower computational complexity.

In Figure 7, a standard large nuclear convolution can be decomposed into deep convolution, deep dilation convolution and point convolution.

Figure 7

Large 5×5 convolution decomposed into 3×3 deep convolution, 3×3 deep dilation convolution, and point convolution

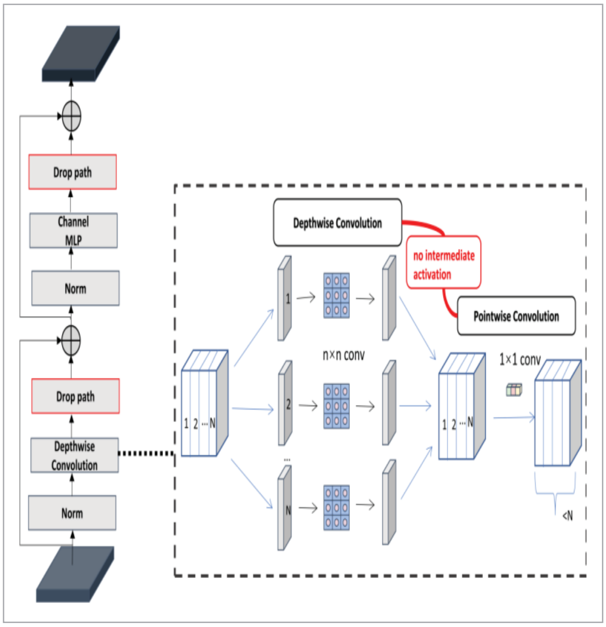


The LKA module effectively emulates the spatial and channel dependencies existing between various regions of an image, enabling adaptive adjustment of attention weights based on input features. This module processes high-resolution images by selectively aggregating information from distinct channels and spatial positions. Selective aggregation does not necessitate matrix multiplication across all feature vectors, thus sidestepping the computational expenses linked with high-resolution image processing. Consequently, it emerges as an effective model suitable for high resolution and multi-scale complex feature target detection. Incorporating the LKA module into the proposed PVF module permits the extraction of local feature information and the aggregation of long-distance dependencies through the utilization of spatial background information. This, in turn, provides a more refined attention guidance mechanism for the target detection model.

– **Visual Mlp**

Figure 8

Vision MLP module



Vision MLP is a new module tailored for computer vision applications. The model architecture is illustrated in Figure 8. It comprises an optimized depth separable convolution and a channel MLP. Within this setup, the attention-based module is replaced by the channel MLP, which takes the output derived from the depth separable convolution as input. The optimized deep convolution in Vision MLP processes the input feature graph, generating a set of channel features fed into the channel MLP. To enhance model generalization and robustness, drop path regularization is applied after deep separable convolution and channel MLP.

Based on research by Chollet et al. [3], the use of nonlinear activation functions in deep separable convolution results in the loss of deep features. While spatial convolution entails linear operation via the convolution kernel on input images, the input information often lacks linear separability. Therefore, applying nonlinear activation functions to spatial convolution enhances model expressiveness. However, both separable convolution and point convolution within deep separable convolution involve nonlinear operations. Inserting additional nonlinear activation functions causes feature information loss and overfitting. Therefore, the nonlinear activation function between separable convolution and point convolution in depth separable convolution is omitted. YOLOX employs the silu activation function expressed as follows, in Equation (9):

$$f(x)=x*\frac{1}{(1+\exp(-x))}. \qquad (9)$$

To improve the generalization and robustness, drop path regularization [6] is introduced after deep separable convolution and channel MLP. During training, it randomly eliminates entire paths within the network to effectively prevent overfitting. This path elimination operation can be perceived as the structural transformation of the network model. Discarded paths are randomly reselected in the ensuing training iterations, rendering the network structure more diverse [24].

To capture long-term dependencies and global relationships, as well as spatial relationships among different images, the channel MLP serves as the output of this module. The channel MLP adaptively processes image size variations. Each color channel corresponds to an MLP layer, enabling feature extraction. The channel MLP treats each channel in the input feature graph as an independent vector, amalgamating and transforming them through the fully connected layer to yield new feature representations. The output of a single vector x_i can be expressed as Equation (10):

$$\mathrm{MLP}(x_1)=\mathrm{Leaky\ Relu}\left(W_1^i * X_1+b_1^i\right), \quad (10)$$

where Leaky Relu represents the activation function, W_1^i is the learnable weight, and b_1^i denotes the bias. The output of this module can be obtained by combining the acquired *i* output vectors as Equation (11):

$$\mathrm{Visual\ Mlp}_{\ out}=\mathrm{Drp}\left[\mathrm{cat}\left(\mathrm{MLP}(x_1),\mathrm{MLP}(x_2),\ldots \mathrm{MLP}(x_i)\right)\right], \quad (11)$$

where Drp signifies drop path.

3.2.3. Gradient Centralization Optimization

Modifying the gradient optimization algorithm can impact model performance [23]. Unlike conventional Batch Gradient Descent (BGD), Stochastic Gradient Descent (SGD) uses only one sample at a time to calculate the gradient and update the model parameters. Therefore, it offers advantages like reduced computational costs and storage requirements. In addition, SGD tends to converge more rapidly to locally optimal solutions and can manage non-convex objective functions. SGD is crucial in training autoencoders to learn basic signal properties from normal signals and minimize reconstruction errors. Optimizations using SGD and an entropy-based objective function during the training phase of the model can lead to better convergence results for the trained model [26]. However, using the SGD optimizer cannot resolve the gradient explosion problem due to network complexity. When the model backpropagates gradients, network updates with multiple hidden layers can become significantly slower than those with a single hidden layer, particularly for complex network architectures like YOLOX [11]. To address these challenges and taking inspiration from Yong et al., this paper introduces the innovative use of the gradient centralization algorithm to optimize YOLOX.

Gradient Centralization is obtained by removing the gradient average of the weight vector from the standard gradient. For the standard convolution layer, it is assumed that the gradient ∇P(Wi) of the weight vector has been obtained by back propagation. Then the weight vector gradient $\nabla_{GC}P(Wi)$ of the gradient concentration algorithm can be defined as Equation (12):

$$\nabla P_{GC}(W_i)=\nabla P(W_i)-\mu, \quad (12)$$

where μ is the gradient mean of the weight vector. It can be defined as Equation (13):

$$\mu=\frac{1}{k}\sum_{j=1}^{k} \nabla P\left(W_{i,j}\right)\ \ (j=1,2,\ldots,N). \quad (13)$$

The steps for applying the algorithm to SGD are shown in the Algorithm 1.

Algorithm1: Stochastic gradient descent using Gradient Centralization (SGDGC)

```
Input: initial weight $W^0$, initial momentum $m^0$, momentum factor $\beta$, initial gradient $g^0$, centralization gradient $g^0_{GC}$, average value $mg^0$.
start:
1. Initialisation $i=0$
2. $g^{\,i}_{GC}= g^i - mg^i$
3. $m^i = \beta m^{i-1} + (1 - \beta)g^{\,i}_{GC}$
4. $W^{i+1} = W^i - m^i$
5. $i=i+1$
While step 1
```

In the experiment, SGDGC is applied to the convolutional and fully connected layers in the YOLOX backbone network due to their output tensor dimension typically exceeding 3, which can potentially lead to gradient explosion.

4. Experiment

4.1. Datasets

This experiment mainly uses two types of datasets: 1) the TT100K dataset, published by Tsinghua and Tencent; 2) the CCTSDB dataset, released by Changsha University of Technology. The CCTSDB dataset includes images that have undergone data enhancement techniques, thereby expanding the scope and quality of the available data. By combining these datasets, this paper aims to address a diverse spectrum of traffic sign detection challenges, providing a more comprehensive evaluation of algorithm performance. The TT100K dataset comprises 10,000 high-resolution traffic sign images captured in real-world environments. These images encompass a total of 30,000 instances of traffic signs with a resolution of 2048×2048. The dataset encompasses 128 distinct traffic sign categories, including but not limited to speed limit signs, prohibition signs, and road signs. To optimize network learning and mitigate the risk of overfitting, traffic sign classes with fewer than 100 instances were removed, ultimately retaining 45 classes as shown in Figure 9. It is noteworthy that the absence of night time traffic sign images in TT100K reduces the generalization ability and robustness of the model. To compensate for this limitation, the experiment uses CCTSDB dataset, a trichotomous dataset classified by meaning as warning, mandatory, and prohibitory, as depicted in Figure 10. This dataset comprises 500 images of traffic signs captured under low-light conditions, enhancing the capacity of the model to detect signs in dark environments. To increase the number of traffic sign images captured in low-light conditions, *imgaug* was used to simulate images under various conditions such as snow, rain, and fog, as illustrated in Figure 11. The images result-

Figure 9

The number of instances corresponding to each category in the TT100K dataset

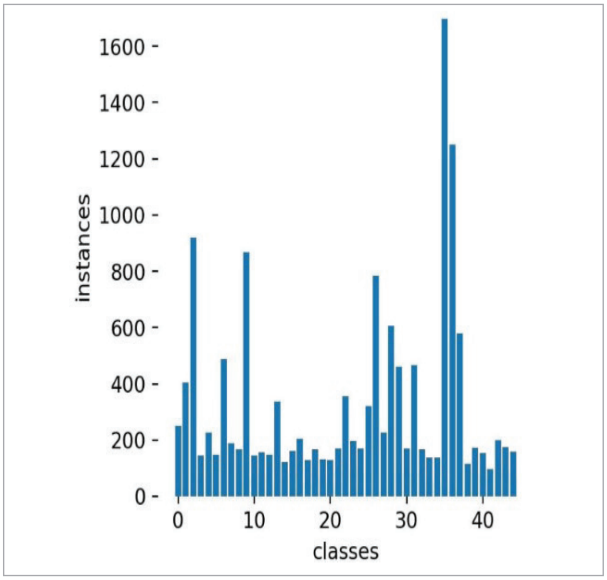


Figure 10

Traffic sign classification map of CCTSDB dataset

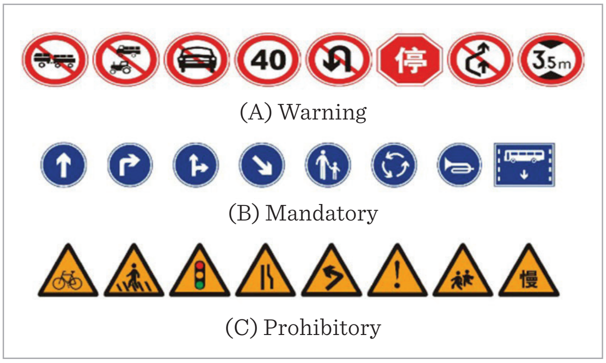
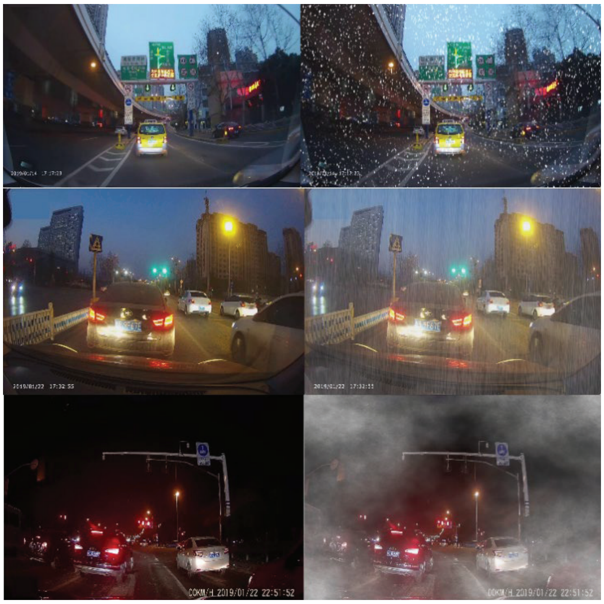


Figure 11

Original images and data augmentation images. The original images are shown on the left and the data augmentation images are shown on the right



ing from data augmentation maintain the same traffic signs as the original images. Through judicious data augmentation, this experiment seeks to bolster the robustness of the model against inclement weather conditions, as later experiments corroborate.

4.2. Experimental Setting

The present study was carried out on an Ubuntu 18.04 operating system with hardware specifications including an i7-11700 CPU, 80 GB of RAM, and a 3090 graphics card with 24 GB of RAM. YOLOX-S was selected as the baseline model for this experiment due to its relatively modest parameter count, improved accuracy compared to other versions of YOLOX, and its suitability for industrial deployment. The experimental settings are as follows: the input image resolution was set to 640×640 pixels. Given the substantial number of images in the dataset, a batch size of 12 was utilized during the training process. Initial learning rate of 0.01 and training momentum of 0.9 were specified, and the training process spanned 200 iterations. Both model width factor and depth factor were set to 1. The experiment incorporated a rotation angle of 10 and a translation range of 0.1. To comprehensively represent model training results, the validation of the model occurred every ten rounds.

4.3. Evaluation Indicator

Experiment use the average detection accuracy (mAP@0.5) detection speed (fps) model size (Model) as the evaluation metric for the model in this experiment. Given that the focus of this experimental model is on traffic sign detection, the accuracy metrics employed are well-suited to effectively assess the recognition performance of the model. These metrics include recognition accuracy and localization accuracy, as well as inference performance. Such metrics are crucial in evaluating the ability of the model to identify and accurately locate objects within the vehicle environment.mAP@0.5 indicates the average accuracy when the IOU threshold is 0.5. Typically, the higher the value of mAP, the better the model is at detecting traffic signs. Whereas fps represents the number of frames per second that the model can handle, when fps is greater than 30, it can be considered as real-time monitoring. In order to illustrate more clearly the detection effectiveness of each model for three different sizes of targets, large, small and medium, the experiments use APl, APm and APs to represent the detection results for large targets (pixel area >96×96), medium targets (pixel area >32×32 and <96×96) and small targets (pixel area <32×32), respectively.

Experiment use the area below the P-R curve to represent the value of mAP, which is a curve with (Precision) as the vertical axis and Recall (Recall) as the horizontal axis, as calculated by the following Equations (14)-(15):

$$\text{Precision}=\frac{TP}{TP+FP} \quad (14)$$

$$\text{Recall}=\frac{TP}{TP+FN} \quad (15)$$

TP is the number of true cases, FP is the number of false positive cases and FN is the number of false negative cases. Precision indicates the number of positive samples in the predicted sample as a proportion of all actual positive samples, and Recall represents the number of positive samples in the predicted sample as a proportion of all actual positive samples.

4.4. Experimental Results and Analysis

To demonstrate the effectiveness of the proposed method, experiments were conducted on the TT100K dataset, and competing algorithms included Efficientdet, Improved Faster R-CNN [6], YOLOv6 [12] and Improved YOLOv3 [21]. The evaluation of the comprehensive performance of the model employs mAP@0.5, fps, and model size, as presented in Table 2. In comparison to other competitive models, the improved algorithm achieves the best trade-off between accuracy and detection speed. Although Improved Faster R-CNN achieves the highest average detection accuracy, its two-stage detection approach is unsuitable for real-time traffic sign detection due to its slower detection speed. In contrast with advanced one-stage object detection algorithms such as YOLOv6-s and M-Yolo, though the proposed algorithm yields a larger model, it excels in the detection of small and large objects. Moreover, its higher detection speed further accentuates its advantage. Experimental results of PVF-YOLO on the TT100K dataset are illustrated in Figure 12. For most categories, the model accurately detected the corresponding category names. While the proposed model exhibited some detection errors for small targets such as “p10”, “pl30”, and “pl40”, overall, the detection outcomes for all targets were satisfactory.

Table 2

Comparison of experimental results on TT100K dataset

Method	Model	mAP@0.5	APl	APm	APs	Fps
Efficientdet	15M	0.693	0.607	——	——	26
Improved Faster R-CNN	120M	0.845	0.836	0.844	0.821	8.4
YOLOv6	57M	0.659	0.671	0.642	0.548	70
Improved YOLOv3	14M	0.756	0.782	0.736	0.748	31
Pvf-YOLO	**86M**	**0.807**	**0.809**	**0.818**	**0.781**	**73**

Figure 12

Full category confusion matrix

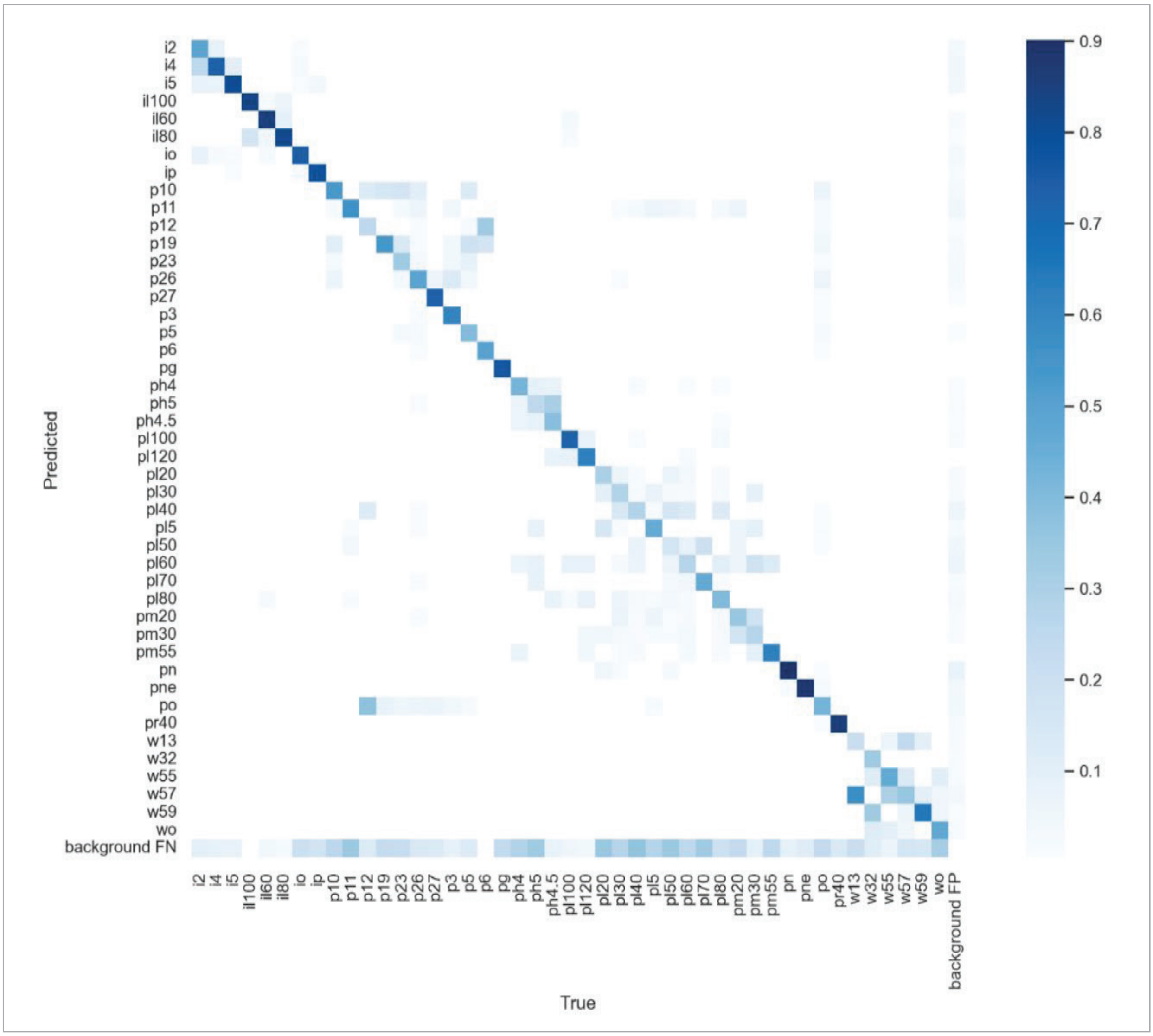
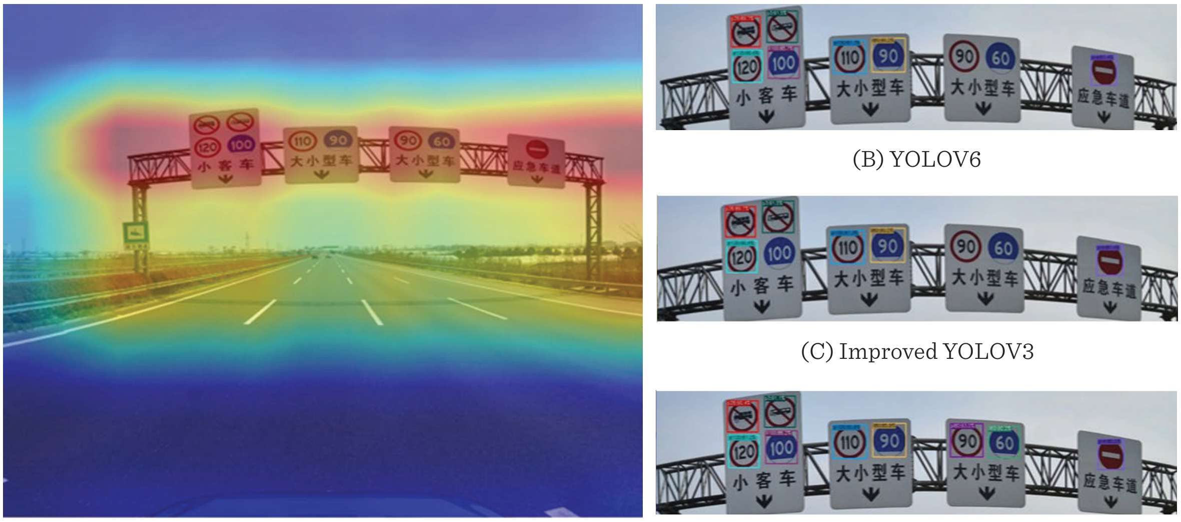


Figure 13



(B) YOLOV6

(C) Improved YOLOV3

(A) Heatmap of the model's region of interest

(D) PVF-YOLO

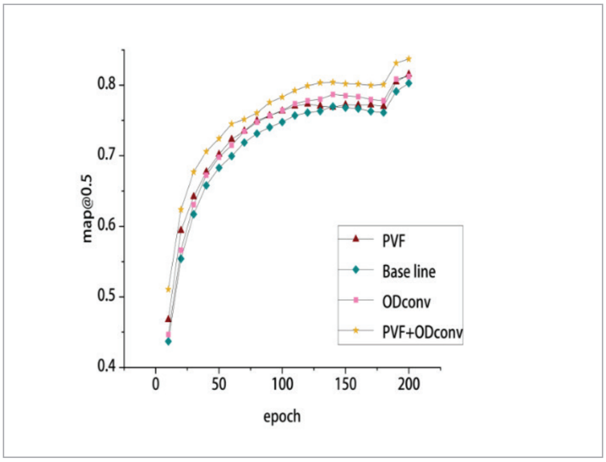
To evaluate the effectiveness of the PVF-YOLO algorithm, real-world traffic signs were employed to test the capabilities of the model, as depicted in Figure 13. The captured images encompass both large and small traffic signs, effectively showcasing the detection ability of the proposed model. The experiment first generated the model's regions of interest using Grad-CAM. The detection was then performed using YOLOv6, Improved YOLOv3, and PVF-YOLO, respectively. The observation reveals that the other two models displayed instances of missed detection and false positives for traffic signs of diverse shapes. In contrast, the detection result of PVF-YOLO aligned seamlessly with the model's regions of interest, successfully and accurately detecting each type of traffic signs.

In this section, this paper conducts a series of ablation experiments to investigate the impact of different components and strategies on the performance of the proposed object detection framework. The baseline configuration comprises YOLOX-S, utilizing SGD as the optimizer. The remaining hyperparameters of the model remain the same as in Section 4.2. This baseline configuration serves as the starting point for the ablation study. The experimental results are presented in Table 3. First, when SGDGC is employed as the optimizer, it improves both the mAP@0.5 and fps of the model. Specifically, by subtracting the mean value from the gradient values, we effectively enhance the gradient propagation within the convolutional layer. The experiment visualizes the gradient values of the

Table 3
Ablation experiments performed on CCTSDB

YOLOX-S (base line)	SGDGC	ODConv	PVF block	mAP@0.5	Model	Fps
√	×	×	×	0.802	69M	59
√	√	×	×	0.810	69M	64
√	√	√	×	0.822	73M	64
√	√	√	√	0.848	80M	73

Figure 14
Ablation experiments



convolutional layers as depicted in Figure 16. From the figure, it is evident that the generation of outliers is suppressed when SGDDC is used as the optimizer. Additionally, the extreme gradient values generated by backpropagation are reduced, diminishing the likelihood of exploding gradients. The training visualization parameters of PVF-YOLO are illustrated in Figure 15. The displayed metrics include mAP@0.5, precision, recall, and classification loss. Considering the varying sizes of traffic signs within the experimental dataset, multiple signs of different dimensions might appear in the same scene. Specifically, the sign information contained in each frame of the video may differ due to the changing speed of the car. This results in a non-uniform feature scale of the target to be detected within adjacent frames in the dataset. Only by extracting both deep and shallow features can the complete evolution of feature information be perceived. The experiments demonstrate the effectiveness of the proposed model, and this comparison is presented in Figure 14 which illustrates the difference in accuracy between using the PVF block or ODConvs alone and using them in conjunction. It is worth noting that since YOLOX-S defaults to turning off data augmentation for the last 15 epochs, a distinct inflection point becomes apparent.

Figure 15
The model uses each parameter curve trained by CCTSDB for the dataset

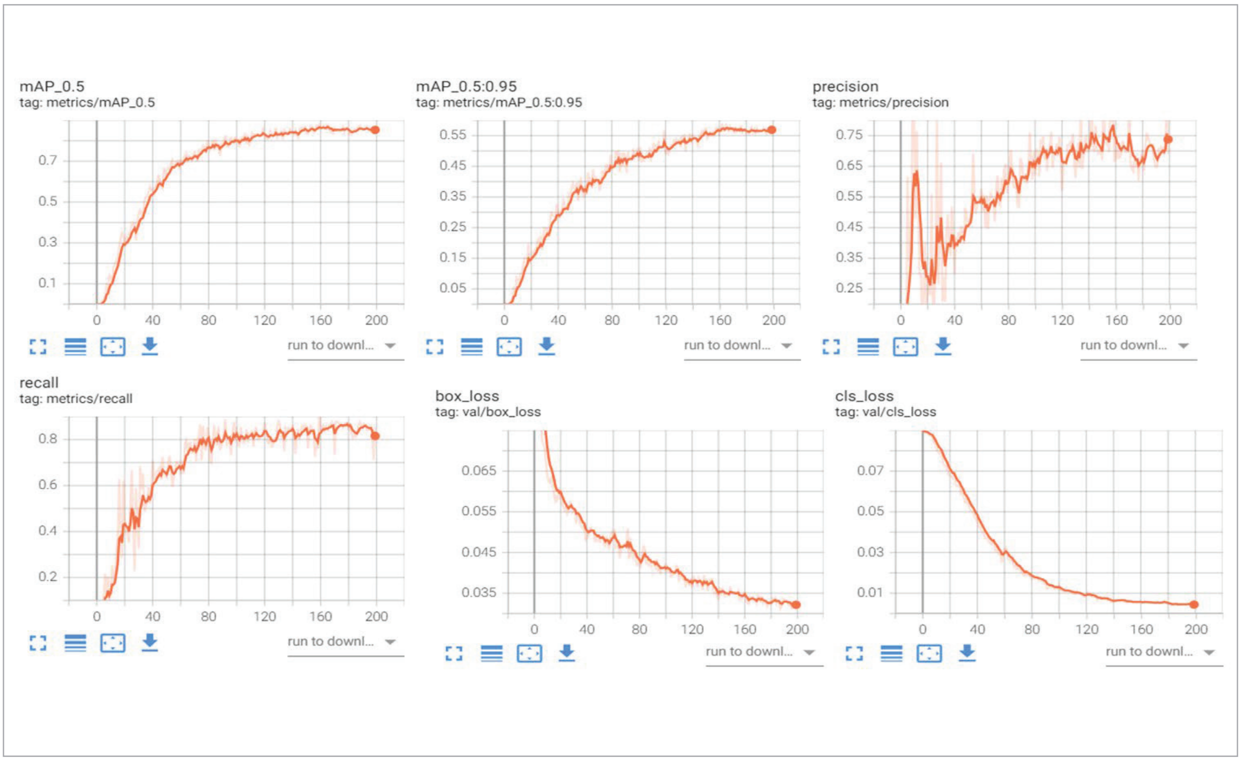
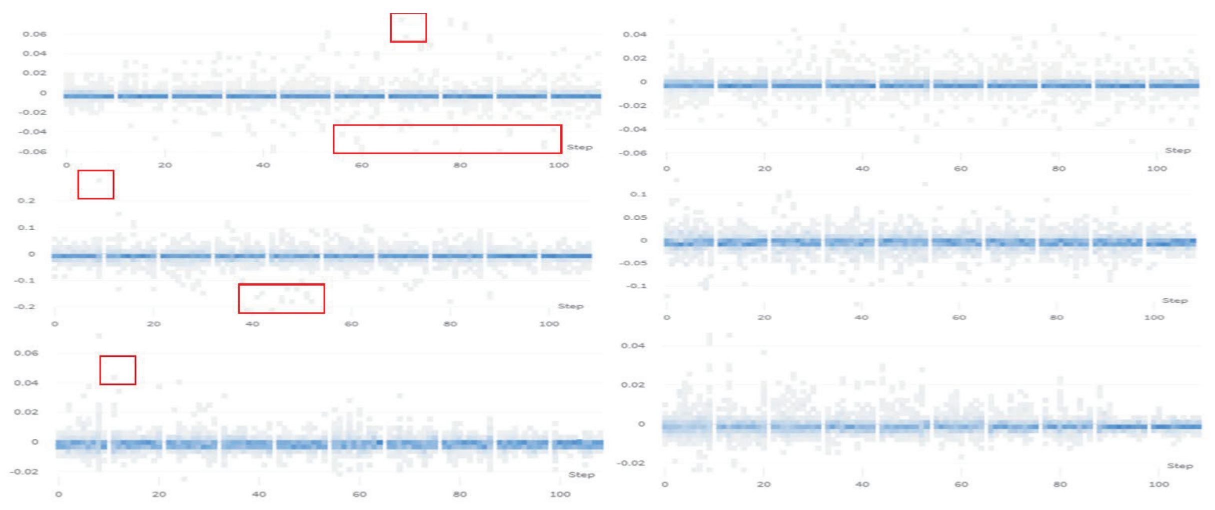


Figure 16

Gradient data maps generated by the model at conv0, conv1, and conv2 layers by backpropagation

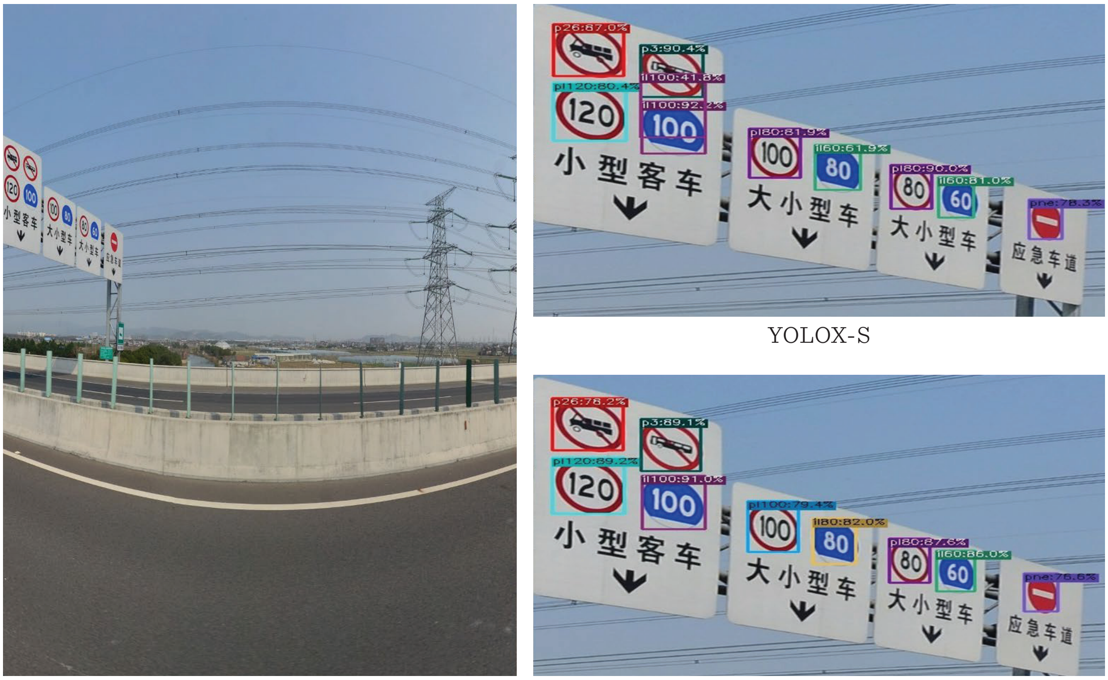


(A) Backpropagated gradients produced using the SGD optimizer

(A) Backpropagated gradients produced using the SGDGS optimizer

Figure 17

Daytime detection results

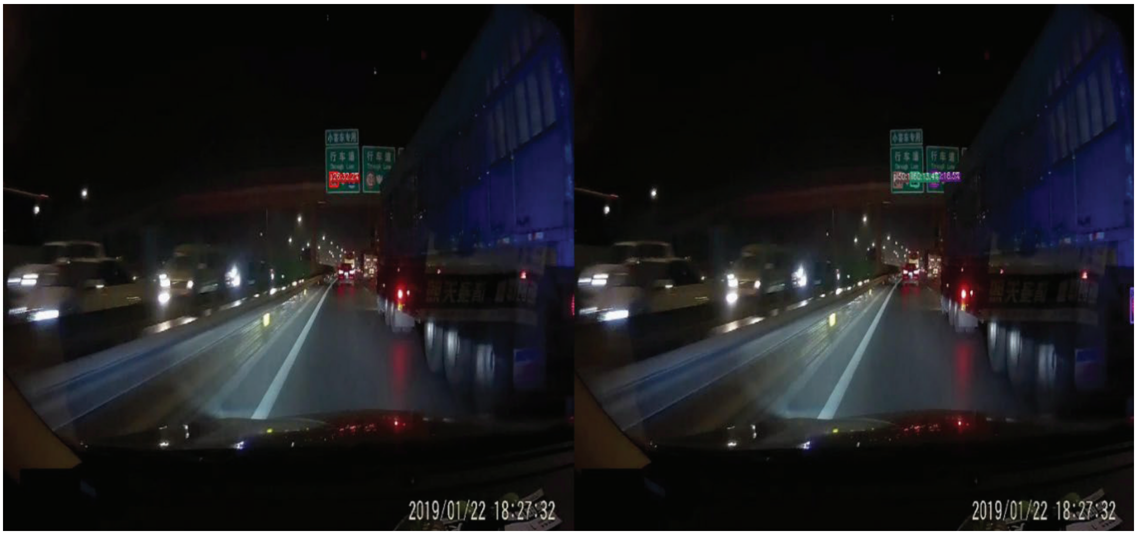


YOLOX-S

Original image

PVF-YOLO

Figure 18
Night time detection effect



YOLOX-S PVF-YOLO

The proposed model is applied in practice to detect results as shown in Figures 17-18. When compared with the original model, the proposed model successfully identifies small, medium, and large traffic signs in real traffic scenes with higher recognition accuracy and almost no instances of missing or false detection. It also performs well in detecting targets against a night time background. These results visually showcase the advanced capabilities of the proposed model.

5. Conclusion

To address the issue of scale changes in traffic signs during vehicle movement, this paper proposes the PVF-YOLO (parallel visual feature) model for extracting both deep and shallow features.

To optimize the YOLOX backbone network, the ODconv replaces the original convolution, and the shallow feature layer is integrated with the neck of the model. The objective is to extract more critical information and enhance model performance.

Using these optimization techniques, the YOLOX backbone network achieves higher accuracy rates in target detection tasks. In addition, a PVF module is introduced to capture deep image feature maps. This module incorporates a large kernel attention module and a visual multilayer perceptron to effectively fuse local and global image features. This approach enables the parallel extraction of deep image feature maps, leading to improved image recognition accuracy.

This paper examines the impact of different model structures on task performance and compares the performance of convolutional layer models with varying depths against the original model. Experiments demonstrate a 2% increase in recognition accuracy for models with both shallow and deep convolutional layers. In addition, the influence of various optimizers on model performance is explored through the integration of gradient centralization into the original optimizer. This integrated approach enhances the focus of the optimizer on gradient information and facilitates more efficient optimization of model parameters. Consequently, this integration not only enhances model accuracy, but also accelerates the model iteration process. Using an enhanced dataset, experimental results indicate that compared to the original YOLOX model, the proposed model achieves a 4% increase in accuracy and a 14 fps improvement in recognition speed. When compared to other mainstream detection models, this proposed model demonstrates significantly improved detection accuracy for traffic targets of varying scales. These results confirm the efficacy of the proposed model in effectively addressing the challenges associated with traffic target detection tasks.

Appendix A

The dataset used in this paper can be found at:

TT100K
https://cg.cs.tsinghua.edu.cn/traffic-sign/

CCTSDB
https://github.com/csust7zhangjm/CCTSDB2021

Acknowledgements

Tian-Wei Shi has supported by the Natural Science Foundation project of Liaoning Province (2021-KF-12-06), Department of Education of Liaoning Province (2020FWDF01) and Science and technology Project of Liaoning Province (2022JH1/10400005).

References

1. Aghdam, H. H., Heravi, E. J., Puig, D. Toward an optimal Convolutional Neural Network for Traf-fic Sign Recognition. Eighth International Con-ference on Machine Vision (ICMV 2015), 9875. SPIE, 2015, 108-112. https://doi.org/10.1117/12.2228582
2. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z. Dynamic Convolution: Attention Over Convolution Kernels. 2020 IEEE/CVF Confer-ence on Computer Vision and Pattern Recogni-tion (CVPR). IEEE, Seattle, WA, USA, 2020, 11027-11036. https://doi.org/10.1109/CVPR42600.2020.01104
3. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017, 1800-1807. https://doi.org/10.1109/CVPR.2017.195
4. Cireşan, D., Meier, U., Masci, J., Schmidhuber, J. Multi-column Deep Neural Network for Traffic Sign Classification. Neural Networks, 2012, 32, 333-338. https://doi.org/10.1016/j.neunet.2012.02.023
5. Di Stefano, L., Mattoccia, S., Tombari, F. ZNCC-based Template Matching Using Bounded Par-tial Correlation. Pattern Recognition Letters, 2005, 26(14), 2129-2134. https://doi.org/10.1016/j.patrec.2005.03.022
6. Gao, X., Chen, L., Wang, K., Xiong, X., Wang, H., Li, Y. Improved Traffic Sign Detection Algo-rithm Based on Faster R-CNN. Applied Scienc-es, 2022, 12(18), 8948. https://doi.org/10.3390/app12188948
7. Guo, M. H., Lu, C. Z., Liu, Z. N., Cheng, M. M., Hu, S. M. Visual Attention Network. Computa-tional Visual Media, 2023, 9(4), 733-752. https://doi.org/10.1007/s41095-023-0364-2
8. Guo, S., Yang, X. Fast Recognition Algorithm for Static Traffic Sign Information. Open Phys-ics, 2018, 16(1), 1149-1156. https://doi.org/10.1515/phys-2018-0135
9. Han, Y., Liu, Y., Paz, D., Christensen, H. Auto-calibration Method Using Stop Signs for Urban Autonomous Driving Applications. 2021 IEEE International Conference on Robotics and Au-tomation (ICRA). IEEE, Xi'an, China, 2021, 13179-13185. https://doi.org/10.1109/ICRA48506.2021.9561909
10. Huang, G., Sun, Y., Liu, Z., Sedra, D., Wein-berger, K. Q. Deep Networks with Stochastic Depth. Computer Vision - ECCV 2016 (Leibe, B., Matas, J., Sebe, N. and Welling, M., eds). Lecture Notes in Computer Science, 9908. Springer International Publishing, Cham, 2016, 646-661. https://doi.org/10.1007/978-3-319-46493-0_39
11. Yang, Z. Intelligent Recognition of Traffic Signs Based on Improved YOLOv3 Algorithm. Mobile Information Systems, 2022, 7877032. https://doi.org/10.1155/2022/7877032
12. Yong, H., Huang, J., Hua, X., Zhang, L. Gradient Centralization: A New Optimization Technique for Deep Neural Networks. Computer Vision - ECCV 2020 (Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.-M., eds). Lecture Notes in Com-puter Science, 12346. Springer International Publishing, Cham, 2020, 635-652. https://doi.org/10.1007/978-3-030-58452-8_37
13. Yu, Y., Liu, F. Effective Neural Network Train-ing With a New Weighting Mechanism-Based Optimization Algorithm. IEEE Access, 2019, 7, 72403-72410. https://doi.org/10.1109/ACCESS.2019.2919987
14. Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., Yan, S. MetaFormer is Actually What You Need for Vision. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, New Orleans, LA, USA, 2022, 10809-10819. https://doi.org/10.1109/CVPR52688.2022.01055
15. Laidlow, T., Czarnowski, J., Leutenegger, S. DeepFusion: Real-Time Dense 3D Reconstruc-tion for Mo-

nocular SLAM using Single-View Depth and Gradient Predictions. 2019 Interna-tional Conference on Robotics and Automation (ICRA). IEEE, Montreal, QC, Canada, 2019, 4068-4074. https://doi.org/10.1109/ICRA.2019.8793527

16. Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., Xie, S. A ConvNet for the 2020s. 2022 IEEE/CVF Conference on Computer Vi-sion and Pattern Recognition (CVPR). IEEE, New Orleans, LA, USA, 2022, 11966-11976. https://doi.org/10.1109/CVPR52688.2022.01167

17. Qian, J., Lin, J., Bai, D., Xu, R., Lin, H. Omni-Dimensional Dynamic Convolution Meets Bot-tleneck Transformer: A Novel Improved High Accuracy Forest Fire Smoke Detection Model. Forests, 2023, 14(4), 838. https://doi.org/10.3390/f14040838

18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Ob-ject Detection. 2016 IEEE Conference on Com-puter Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, NV, USA, 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

19. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6), 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

20. Saponara, S., Elhanashi, A., Zheng, Q. Develop-ing a Real-time Social Distancing Detection System Based on YOLOv4-tiny and Bird-Eye View for COVID-19. Journal of Real-Time Im-age Processing, 2022, 19(3), 551-563. https://doi.org/10.1007/s11554-022-01203-5

21. Selvaraju, R. R., Cogswell, M., Das, A., Vedan-tam, R., Parikh, D., Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradi-ent-Based Localization. 2017 IEEE Internation-al Conference on Computer Vision (ICCV). IEEE, Venice, 2017, 618-626. https://doi.org/10.1109/ICCV.2017.74

22. Tan, M., Pang, R., Le, Q. V. EfficientDet: Scala-ble and Efficient Object Detection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Seattle, WA, USA, 2020, 10778-10787. https://doi.org/10.1109/CVPR42600.2020.01079

23. Tang, J., Li, Q. Fast Template Matching Algo-rithm: Fast Template Matching Algorithm. Jour-nal of Computer Applications, 2010, 30(6), 1559-1561. https://doi.org/10.3724/SP.J.1087.2010.01559

24. Wang, J., Chen, Y., Dong, Z., Gao, M. Improved YOLOv5 Network for Real-Time Multi-Scale Traffic Sign Detection. Neural Computing and Applications, 2023, 35(10), 7853-7865. https://doi.org/10.1007/s00521-022-08077-5

25. Zhang, J., Zou, X., Kuang, L.-D., Wang, J., Sher-ratt, R. S., Yu, X. CCTSDB 2021: A More Com-prehensive Traffic Sign Detection Benchmark. Human-centric Computing and Information Sciences, 2022, 12(0), 289-306. https://doi.org/10.22967/HCIS.2022.12.023

26. Zheng, Q., Zhao, P., Wang, H., Elhanashi, A., Sa-ponara, S. Fine-Grained Modulation Classifica-tion Using Multi-Scale Radio Transformer With Dual-Channel Representation. IEEE Communi-cations Letters, 2022, 26(6), 1298-1302. https://doi.org/10.1109/LCOMM.2022.3145647

27. Zheng, Q., Zhao, P., Zhang, D., Wang, H. MR-DCAE: Manifold Regularization-based Deep Convolutional Autoencoder for Unauthorized Broadcasting Identification. International Jour-nal of Intelligent Systems, 2021, 36(12), 7204-7238. https://doi.org/10.1002/int.22586

28. Zhu, X., Lyu, S., Wang, X., Zhao, Q. TPH-YOLOv5: Improved YOLOv5 Based on Trans-former Prediction Head for Object Detection on Drone-captured Scenarios. 2021 IEEE/CVF In-ternational Conference on Computer Vision Workshops (ICCVW). IEEE, Montreal, BC, Canada, 2021, 2778-2788. https://doi.org/10.1109/ICCVW54120.2021.00312