


Article

PSG-Yolov5: A Paradigm for Traffic Sign Detection and Recognition Algorithm Based on Deep Learning

Jie Hu *, Zhanbin Wang , Minjie Chang, Lihao Xie, Wencai Xu and Nan Chen

College of Automotive Engineering, Wuhan University of Technology, Wuhan 430070, China

* Correspondence: auto_hj@163.com

Abstract: With the gradual popularization of autonomous driving technology, how to obtain traffic sign information efficiently and accurately is very important for subsequent decision-making and planning tasks. Traffic sign detection and recognition (TSDR) algorithms include color-based, shape-based, and machine learning based. However, the algorithms mentioned above are insufficient for traffic sign detection tasks in complex environments. In this paper, we propose a traffic sign detection and recognition paradigm based on deep learning algorithms. First, to solve the problem of insufficient spatial information in high-level features of small traffic signs, the parallel deformable convolution module (PDCM) is proposed in this paper. PDCM adaptively acquires the corresponding receptive field preserving the integrity of the abstract information through symmetrical branches thereby improving the feature extraction capability. Simultaneously, we propose sub-pixel convolution attention module (SCAM) based on the attention mechanism to alleviate the influence of scale distribution. Distinguishing itself from other feature fusion, our proposed method can better focus on the information of scale distribution through the attention module. Eventually, we introduce GSConv to further reduce the computational complexity of our proposed algorithm, better satisfying industrial application. Experimental results demonstrate that our proposed methods can effectively improve performance, both in detection accuracy and mAP@0.5. Specifically, when the proposed PDCM, SCAM, and GSConv are applied to the Yolov5, it achieves 89.2% mAP@0.5 in TT100K, which exceeds the benchmark network by 4.9%.

Keywords: traffic sign detection; deep learning; small object detection; multi-scale fusion



Citation: Hu, J.; Wang, Z.; Chang, M.; Xie, L.; Xu, W.; Chen, N. PSG-Yolov5: A Paradigm for Traffic Sign Detection and Recognition Algorithm Based on Deep Learning. *Symmetry* **2022**, *14*, 2262. <https://doi.org/10.3390/sym14112262>

Academic Editors: João Ruivo Paulo, Cristina P. Santos and Gabriel Pires

Received: 27 September 2022

Accepted: 11 October 2022

Published: 28 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

As part of intelligent transportation system, autonomous driving is largely regarded as a promising technology for reducing traveling time, traffic load, and guaranteeing traveling safety, thereby reducing the incidence of traffic accidents. Traffic sign detection has gradually become a fundamental perception task involved in the development of intelligent transportation systems and autonomous vehicles. The research motivation of this paper is to improve the detection accuracy of traffic sign detection task in autonomous driving and empower autonomous driving to be realistic. Traffic signs can generally be divided into warning signs, instruction signs, and prohibition signs, all of which contain rich semantic information. The use of simple geometric shapes and bright colors make it easy for the human eye to acquire traffic sign information; however, the eyes of traffic sign detection systems: vision-based sensors are highly susceptible to the size of traffic signs and other external environmental factors, which affect the driving safety seriously of autonomous vehicles.

The current traffic sign detection technology has the following several deficiencies: first, the size of traffic signs occupies a small proportion of the real road scene, which is difficult for traffic sign detection systems to capture accurate traffic sign information. Taking the traffic sign dataset Tsinghua-Tencent 100k Dataset (TT100k [1]) as an example, the proportion of the pixel range of traffic signs is only 0.2% of the image pixel [2],

illustrated in Figure 1. Second, there exist large-scale or small-scale traffic signs in the same image, difference in scale can easily cause false or missed detections of the detector, furthermore, seriously affecting the detection accuracy. Eventually, the deployment of traffic sign detection technology requires not only high accuracy but a high inference speed to satisfy the real-time detection in complex traffic environments.

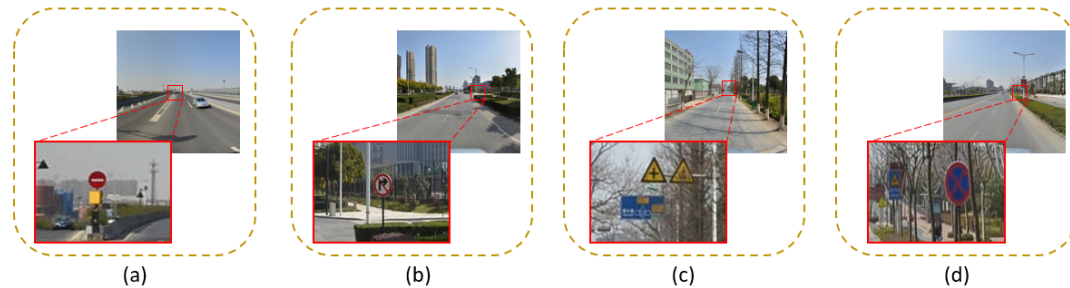


Figure 1. (a–d) Some images from the TT100k dataset, the traffic signs' pixel value occupying only a small fraction of the image.

To alleviate the above problems, we propose a PSG-Yolov5 traffic sign detection algorithm based on Yolov5 (5th version of You Only Look Once [3]). Experiments indicate that our proposed traffic sign detector achieves relatively ideal detection and classification accuracy. The main contributions of this paper are as follows:

- In order to eliminate the problem that most small-scale traffic signs' spatial information can only exist in the shallow network and cannot be transmitted to the deep network through the feature extraction process, we propose a plug-and-play adaptive feature extraction module, parallel deformable convolution module (PDCM), as shown in Section 3.2. The proposed method divides the input features equally in the channel dimension [4], with each branch extracting the features individually. By introducing deformable convolution, the traffic sign detector can improve the feature extraction capability and modeling capability of CNNs, better preserving the spatial information of small-scale traffic signs.
- Inspired by the sub-pixel convolution in [5], we propose SCAM in this paper. Our proposed SCAM can more effectively utilize the rich semantic information of high-level feature maps, better capturing the correlation between adjacent feature layers, filtering out redundant information, and making the output feature map pay more attention to the traffic sign information of the corresponding scale. It can also alleviate the aliasing effect caused by feature fusion [6].
- We build our traffic sign detection model by referring to [7], which further reduces the computational complexity of our proposed algorithm, making our proposed traffic sign detector industrially applicable.
- We optimize our traffic sign detector and integrate it with Robot Operating System (ROS) [8], deploying our proposed traffic sign detector to the Nvidia Jetson Xavier. We provide a concrete reference for industrial applications of traffic sign detection systems on edge devices.

The rest of the article is organized as follows: we present a brief overview of related work and methods concerning traffic sign detection in Section 2. Section 3 provides details of our research. In Section 4 we present the experimental results, and finally, in Section 5 we summarize all the work and provide conclusions.

2. Related Work

Traffic sign detection (TSD), as a branch of target detection, has gradually become a hot research topic in recent years. The research history of traffic sign detection can be broadly divided into traditional algorithms and deep learning algorithms.

2.1. Traditional Algorithms

Traditional TSD can be divided into color-based, shape-based, and machine learning based. Color-based and shape-based detection techniques mainly utilize specific image colors and shapes to manually extract features, such as SIFT [9] (scale invariant feature transform) features, HOG [10] [Histograms of Oriented Gradient] features, these traffic sign detection techniques mainly extract visual information in candidate regions, cropping and extracting traffic signs in the images, and match marker signs by templates. Takaki Masanari [9], and others use SIFT method to detect traffic signs. However, color-based and shape-based methods are susceptible to weather conditions, lighting, and other environmental factors [11]. Machine learning methods are used to extract invariant or similar visual features from traffic signs, detect the traffic signs in the image, then use classification algorithms to classify them and hence understand the semantic information contained in the traffic signs. Classical classification algorithms include template matching algorithms and support vector machines (SVMs), random forests, etc. However, after nearly decades of research, the performance of algorithms for traffic sign detection by manually designed features has reached a bottleneck.

2.2. Deep Learning Algorithms

Deep learning came to researchers' attention in 2012 with the introduction of AlexNet [12], a convolutional neural network (CNN) approach to detecting traffic signs. Deep learning algorithms can independently train and learn network models based on labeled object datasets, and along with the development of parallel computing devices, the dataset has been accompanied by further expansion, allowing for further robust algorithms to be applied in the TSD field. The algorithms can be roughly divided into one-stage and two-stage detection algorithms depending on whether candidate frames are generated.

2.2.1. Two-Stage Algorithms

Two-stage detection algorithms mainly include R-CNN [13], Fast-RCNN [14], Faster R-CNN [15], Mask R-CNN [16], and Cascade R-CNN [17]. Due to the complexity of the traffic sign detection task, CAO [18] proposed a multi-scale fusion detection method based on Faster R-CNN and verified the robustness of their algorithm; TANG [2] proposed a feature aggregation network to enhance the feature extraction capability. In general, the accuracy of two-stage detection algorithms is higher than one-stage. Although the above-mentioned algorithms enhance the detection capability of the model to a certain degree. The limitations of their methods are unable to meet the real-time requirements, therefore, improvements for one-stage detection algorithms are very valuable for research.

2.2.2. One-Stage Algorithms

One-stage detection algorithms are represented by SSD [19], YOLO [3], etc. He [20] applied SPP to collect features at different scales learning multi-scale features more comprehensively, which obtained 97% accuracy with an average inference speed of 19.3 ms per frame on the CCTSDB dataset. The M-Yolo traffic sign detection model proposed by Liu [21], achieves an accuracy of 93.5% on the CCTSDB dataset by introducing network structures such as FOCUS [22] and SPPF, etc. The above improvements increase the detection accuracy as much as possible while keeping inference speed, reaching an accuracy sufficient to rival algorithms such as Faster R-CNN with less inference time. In summary, the state-of-the-art Yolo algorithm is very suitable for the recognition of traffic sign detection due to its outstanding real-time performance, generalization ability, and its remarkable performance of fine-grained instances, etc. Although the above researchers' one-stage models had improved the detection accuracy, there is still some room for further improvement in small-scale traffic signs detection and multi-scale feature fusion process.

3.2. Parallel Deformable Convolution Module

As shown in Figure 3, we propose a parallel and symmetrical deformable convolution module, which can enhance the spatial information extraction capability of the backbone network by adaptively changing the shape of convolutional kernel. Because the shallow neural network structure is rich in spatial information, we apply our proposed module to the shallow network to effectively improve the modeling capability of CNNs when capturing small-scale traffic signs. In PDCM, we first equally split the input feature map in the channel dimension, and then independently change the perceptive field size through two identical separate branches. Then, we concatenate the two branches in channel dimension, and finally, we shuffle the resulting feature map to obtain the output feature map. C1 and C2 in Figure 3 represent the number of channels of input and output feature maps respectively, and H, W are the height and width of the tensor, respectively.

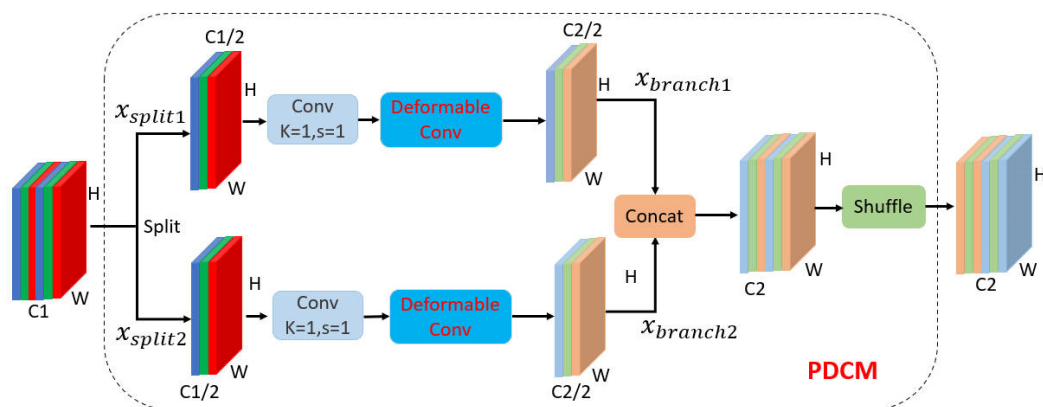


Figure 3. The structure of parallel deformable convolution module. The module consists of a split operation, convolution operation; deformable convolution module; concatenate; and shuffle. It is primarily used to enhance the spatial information extraction and modeling capability of the backbone network by adaptively changing the size of perceptive field.

After feature extraction and concatenation, we obtain a feature map containing rich spatial information about the multi-scale traffic signs, after which we apply a channel shuffling to the feature map. As shown in Figure 3, the feature map with channel size C2 is obtained. The output feature map of the parallel deformable convolution module is shown as Equation (1). $f_{shuffle}$, f_{DCN} , and f_{Conv} represent shuffling channels, Deformable Conv function, and standard convolution function respectively. Our proposed parallel deformable convolution module is very effective in improving the accuracy of traffic sign detection as shown in Section 4.

$$output = f_{shuffle}(cat(f_{DCN}(f_{Conv}(x_{split1})), f_{DCN}(f_{Conv}(x_{split2})))) \tag{1}$$

3.2.1. Identical and Symmetrical Branch Operation

Since the two branches are identical, we only describe one branch in detail. Each branch is composed of convolution module and deformable convolution network (DCN) operation. As illustrated in Figure 3, Conv in every branch represents the standard convolution operation, $k = 1$ and $s = 1$ represent the size of the convolution kernel equaling to 1 and the stride equaling 1 respectively. k and s default to 1 in PDCM unless specially stated. Since CNN models have a fixed geometric structure, such as convolutional layers, pooling layers, etc., which leads to the same perceptive field for all activation units lacking mechanisms to handle geometric variations. Therefore, we refer to deformable convolutional networks [25] and introduce deformable convolution, which is capable of adaptively detecting geometric changes in size, pose, etc. Since small-scale traffic sign recognition and localization are closely related to their shape and pose, DCN is particularly suitable for the recognition of small-scale traffic signs. The Deformable Conv in Figure 3 represents the

standard deformable convolution, which is illustrated in Figure 4. The DCN module first obtains offsets from the input feature maps, the convolution kernel shape is then adaptively changed by the learned offsets.

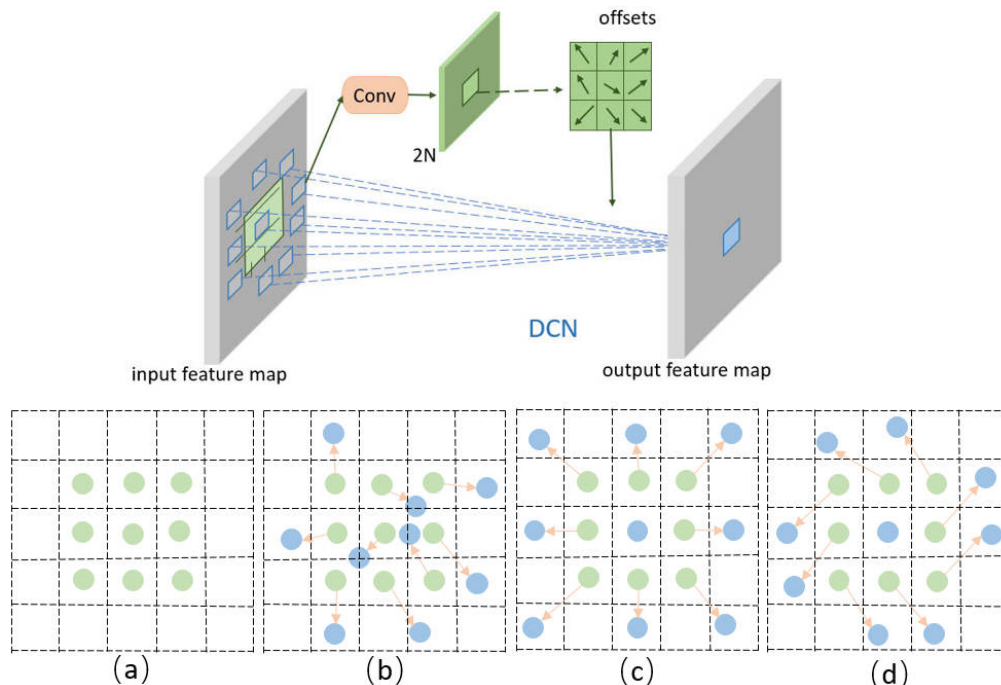


Figure 4. The structure of DCN and introduction of sampling grid [25]. (a) is standard sampling grid, (b–d) represent the deformable sampling grid. The green circles represent the original sampling grid, and the blue circles represent the sampling points after learning offsets.

In PSG-Yolov5, the input feature map which size is $X \in \mathbb{R}^{80 \times 80 \times 256}$. We split the input feature map equally and feed it into two separate branches, whose size is $X \in \mathbb{R}^{80 \times 80 \times 128}$. The operation of each branch after splitting is shown as Equation (2). The f_{Conv} and f_{DCN} represent the Conv operation and DCN operation respectively.

$$x_{branch1} = f_{DCN}(f_{Conv}(x_{split1})) \tag{2}$$

3.2.2. The Introduction of DCN and Conv Module

Deformable convolutional networks proposed deformable convolution that adaptively changes the size of perceptive field to better capture the spatial information. Deformable convolution adds a 2D offset to the regular grid sampling positioning, which allows spontaneous deformation of the sampling grid. The offset is drawn from the previous feature map. The deformation of the sampling grid depends on the input features in a local, density, and adaptive manner. The sampling grid in deformable convolution is indicated in Figure 4. Equation (3) represents the deformable convolution, where P_0 represents each position on the output maps; P_n represents sampling on the input feature map with regular kernel size; Δp_n represents offset for each sample point.

In PDCM, each Conv module consists of three parts, a convolutional-2D layer, a batch normalization-2D layer, and an activation layer. The calculation process is shown in Equation (4), where $\sigma(\cdot)$ and $B(\cdot)$ represent the activation function and Batch Normalization layer [20] respectively.

$$f(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{3}$$

$$f_{Conv} = \sigma(B(x_{split1})) \tag{4}$$

3.3. Sub-Pixel Convolution Attention Module

It is well-known that the information contained in low-level and high-level feature maps are complementary for multi-scale detection tasks, and it is valuable to study how to fully utilize the information at different scales for traffic sign detection tasks.

As shown in Figure 5, we propose the sub-pixel convolution attention module (SCAM). In the traffic sign detection task, the high-level feature maps contain rich semantic information, such as {P3, P4} in Figure 5.

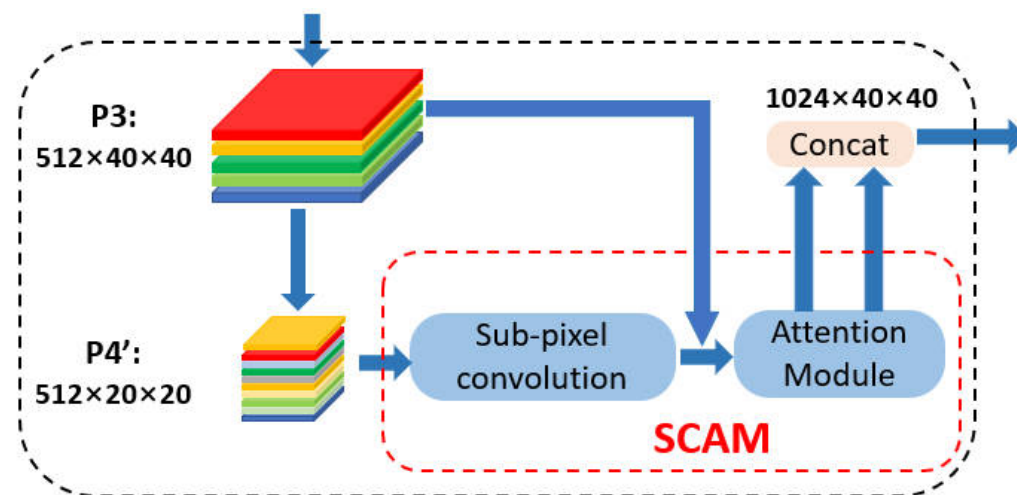


Figure 5. Schematic of SCAM. The high-level feature map P4' (shown in Figure 2) first goes through sub-pixel convolution, and the size change is illustrated in the figure. After this P4' and P3 are input to the attention module at the same time.

Along with the proposed of FPN network [6], which can effectively fuse high-level feature maps, the widely used cross-scale fusion strategy effectively improves detection performance. However, multi-scale feature maps cannot be fused directly due to different scales, interpolation is needed if we wish to effectively utilize the feature maps. We know that direct interpolation-fusion may bring significant aliasing effects, and may seriously affect the classification and localization of traffic sign detection tasks.

Inspired by sub-pixel convolutional networks [5], we introduce sub-pixel convolution to solve the defects caused by nearest neighbor interpolation, and we introduce spatial and channel attention mechanisms to fully utilize the feature layers of different scales. SCAM can adjust the feature layers of different scales to pay more attention to their own corresponding scale information. The overall schematic of the sub-pixel convolution attention module and the changing process of feature dimension is illustrated in Figure 5.

We validate our proposed SCAM in Section 4, and our experimental results show that our approach is effective and brings performance gains with almost negligible computational burden.

3.3.1. Sub-Pixel Convolution

In order to fully utilize the rich semantic information in P4 and P3, we introduce the sub-pixel convolution module by referring to the up-sampling method in super-resolution. As an up-sampling strategy, sub-pixel convolution can modify the input feature’s width and height to turn low-resolution features into high-resolution features through the pixel-shuffle process, which is illustrated in Figure 6, and its mathematical expression is shown in Equation (5). PS represents periodic shuffling operator which turns a tensor of shape $H \times W \times C \cdot r^2$ to $rH \times rW \times C$, r is an up-sampling factor, in our paper r equals to 2, T is input feature map.

$$PS(T)_{X,Y,C} = T_{X/r,Y/r,C \cdot r \cdot \text{mod}(Y,r)+C \cdot \text{mod}(X,r)} \tag{5}$$

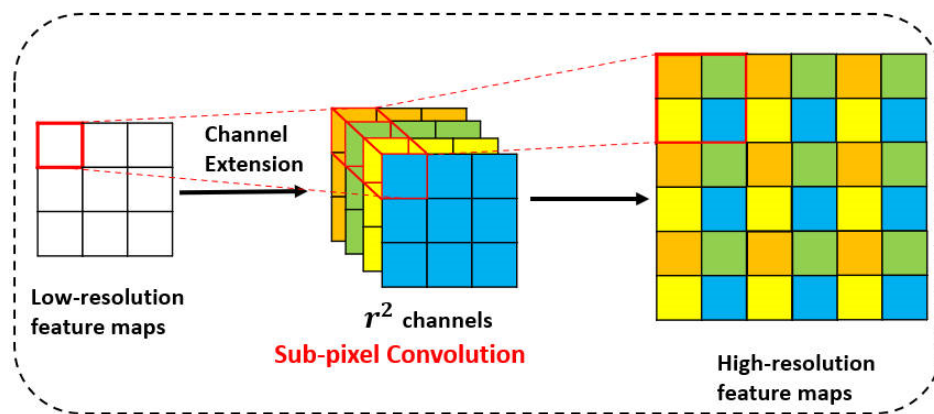


Figure 6. The structure of sub-pixel convolution. The channel of the low-resolution feature map is first expanded by the channel extension (a standard 1×1 convolution) operation, and the high-resolution feature map is obtained by the pixel shuffle operation.

3.3.2. Attention Module

We refer to the CBAM attention mechanism [26], of which the channel attention module and spatial attention module are two submodules respectively.

Channel attention: Keeping the channel dimension unchanged and compresses the spatial dimension. In PSG-Yolov5, the introduction of channel attention can make P3 paying more attention to the medium-scale traffic signs, thus excluding the redundant information. The input feature map first goes through two parallel *MaxPool* layers and *AvgPool* layers, which change the dimension of the feature map from $C \times H \times W$ to $C \times 1 \times 1$, and then goes through the *Share MLP* module, which compresses the channel of the feature map to $1/r$ times of the original, and then expands it to the original channel, and the results of the two branches obtained by the activation function execute the element-wise operation, and then finally by the sigmoid activation function is multiplied with the original feature map. The schematic diagram of channel attention is shown in Figure 7, and the mathematical formula of our channel attention is shown in Equation (6).

$$M_C(P3) = \sigma(MLP(AvgPool(P3)) + MLP(MaxPool(P3))) \tag{6}$$

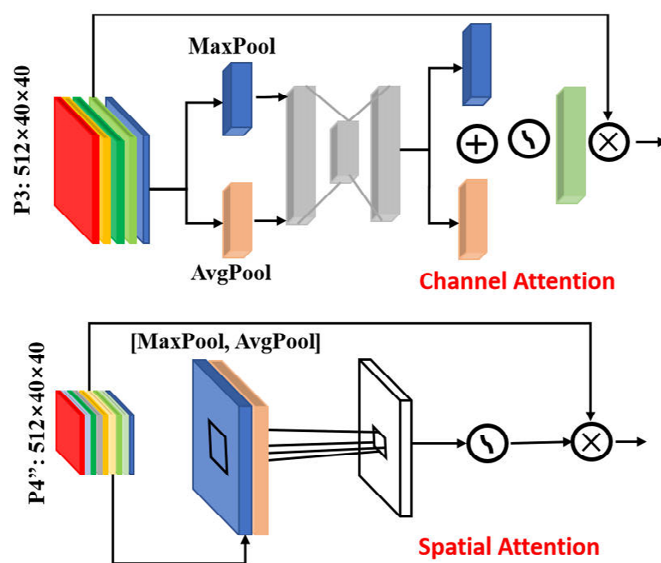


Figure 7. The structure of Spatial and Channel Attention. P3, P4' represent the feature maps captured by the backbone extraction network and P4' after sub-pixel convolution module (shown in Figure 5) respectively. *MaxPool* and *AvgPool* represent the maximum pooling and average pooling.

Spatial attention: Keeping the spatial dimension unchanged, the channel dimension is compressed to render it more focused on the location information of the target. In our traffic sign detection task, the introduction of the spatial attention module can localize traffic signs more efficiently and alleviate the nuisance caused by uneven sample scales. The spatial attention is shown in Figure 7 and its mathematical equation is shown as Equation (7).

$$M_s(P4'') = \sigma(f^{7 \times 7}([AvgPool(P4''); MaxPool(P4'')])) \tag{7}$$

3.4. GSConv Module

Traffic sign detection task in industrial project requires high detection accuracy, in addition, the inference speed is essential. Usually, the higher the number of parameters of the model, the higher detection accuracy will be; however, the pursuit of accuracy is no longer a perfect solution to the traffic sign detection task. To summarize, we refer to GSConv [1] and introduce a lighter convolutional structure to make the number of parameters of our proposed model smaller. We embed the GSConv module into the feature fusion stage so that our model is under a significantly lower number of parameters with slightly lower detection accuracy. We did not use GSConv in the backbone network because it would cause a deeper backbone network layer, and a deeper network would aggravate the resistance to spatial information flow and thus affect the inference speed, which is intolerable in a traffic sign detection system. Figure 8 shows the schematic diagram of the GSConv module. Yolov5: 5th version of Yolo Only Look Once.

GSConv module mainly consists of Conv module, DWConv module, Concat module, and shuffle module, whose mathematical expression is Equation (8), $f_{shuffle}$ means shuffle operation, f_{conv} consists of standard convolution, batch normalization [27] operation and activate function, f_{dsc} represents depth-separable convolution (DSC), batch normalization operation and activate function.

$$X_{out} = f_{shuffle}(cat(f_{conv}(X_{in}), f_{dsc}(f_{conv}(X_{in})))) \tag{8}$$

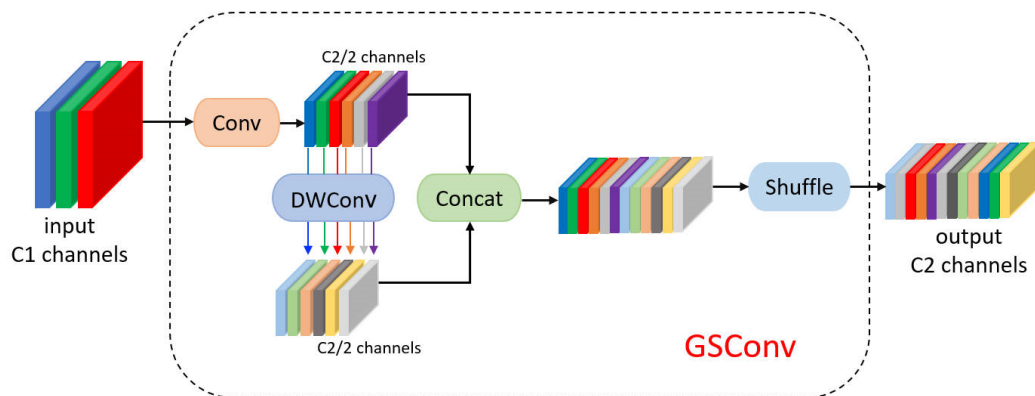


Figure 8. The structure of the GSConv module. DWConv in the figure means depth-separable convolution (DSC). In Figure 8 the “G” represents the GhostNet [28], “S” represents the shuffle operation, and the “Conv” is standard convolution operation.

3.5. The Loss Function of PSG-Yolov5

The loss function of the target detection task is generally composed of bounding boxes regression loss and classification loss. The commonly used calculation indicator of bounding boxes regression loss is the intersection-over-union ratio (IOU [29]), which compares the predicted bounding boxes with the ground truth bounding boxes. The IOU loss function can continuously correct the localization of the predicted bounding boxes through regression. With continuous research on loss functions, many excellent loss functions have been proposed, such as GIOU [30], DIOU [31], CIOU [32], etc. The

mathematical formula expressions of the above four types of loss functions are Equation (9) to Equation (12) respectively.

A and B in the above equations represent the area of ground truth bounding boxes and the area of prediction bounding boxes respectively. C represents the minimum enclosing box of A and B. In Equation (11), d represents the Euclidean distance of the opposite corners of the bounding boxes. $\rho(\cdot)$ represents the calculation expression for Euclidean distance. In Equation (12), the v represents the evaluation metric for evaluating the aspect ratio of ground-truth bounding boxes and predicted bounding boxes, and α represents the indicator of trade-off.

$$Loss_{IOU} = 1 - IOU, IOU = \frac{A \cap B}{A \cup B} \tag{9}$$

$$Loss_{GIOU} = 1 - IOU + \frac{C - (A \cup B)}{C} \tag{10}$$

$$Loss_{DIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{d^2} \tag{11}$$

$$Loss_{CIOU} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{d^2} + \alpha v \tag{12}$$

$$v = \frac{4}{\pi^2} (\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}) 2, \alpha = \frac{v}{(1 - IOU) + v}$$

The IOU loss function has the advantages of scale invariance and non-negativity; however, when the bounding boxes do not intersect, the IOU equaling 0, and the IOU cannot reflect the distance relationship between the predicted bounding box and the ground truth bounding box. GIOU loss function focuses not only on overlapping regions, but also non-overlapping regions. However, the GIOU loss function converges slowly when doing regression tasks. DIOU introduces the Euclidean distance indicator between center point of predicted bounding box and ground truth bounding box, which can accelerate the model convergence, but does not consider the aspect ratio of the bounding box. CIOU, the aspect ratio of the bounding box is increased, which makes up for the deficiency of the DIOU loss function to a certain extent. The PSG-Yolov5 proposed by us adopts the CIOU bounding boxes loss function.

4. Experimental Results and Analysis

In this chapter we introduce the test dataset we used, the related experimental evaluation indexes, the ablation and comparison experimental results and analysis, we also visualize our experimental results in this chapter.

4.1. Traffic Sign Dataset

We utilize the Tsinghua-Tencent 100K Tutorial [1] as the benchmark dataset to test our proposed PSG-Yolov5 algorithm. The dataset comprises 9176 images, including 6105 training images and 3071 test images. The TT100k dataset consists of 227 categories with 2048 × 2048 image resolution, covering scenes with different weather conditions. Table 1 shows a brief description of the dataset.

Table 1. Statistical tables for the TT100k dataset.

Benchmark Datasets	Tsinghua-Tencent 100K
Images	9176 (6105 for training, 3071 for testing)
Categories	227
Resolution	2048 × 2048
GT Boxes	16,527

4.2. Experimental Configuration and Evaluation Metrics

4.2.1. Experimental Configuration

We use the Pytorch framework to build our PSG-Yolov5. Our experiments were conducted on the Ubuntu18.04 operation system with two Nvidia Tesla V100. The hyper-parameter configuration of PSG-Yolov5 is as follows: epoch is 100; batch size is 16; initial learning rate is 0.01 and final learning rate is 0.01; the momentum and weight decay are 0.937 and 0.0005. The optimizer is SGD [33] optimizer. The Nvidia Jetson Xavier's operation system is Linux Ubuntu 18.04.

4.2.2. Experimental Evaluation Metrics

There exist multiple metrics to evaluate algorithms in target detection tasks. Commonly used include, precision(P), which indicates the proportion of predicted positive samples. Recall(R), which indicates the proportion of all predicted positive samples. AP (average-precision), which indicates the average accuracy of different recall points. Mean average-precision(mAP), is the average of AP(average-precision) of multiple categories. The mathematical expressions of the evaluation index mentioned above are illustrated in (13) to (14). *TP* and *TN* represent positive and negative samples with correct prediction respectively, *FP* and *FN* represent positive and negative samples with incorrect prediction respectively. In our experiments, the mAP@0.5 is the average precision of the traffic sign categories when the accuracy evaluation IOU threshold is set to 0.5.

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \quad (13)$$

$$AP_c = \frac{1}{N_c} \sum_{r_c \in R_c} p(r_c), mAP = \frac{1}{N} \sum (AP_c) \quad (14)$$

4.3. Experimental Results

4.3.1. Ablation Experiments

We evaluate the effectiveness of our proposed individual modules, including the PDCM, SCAM, and GSConv convolutional module on the publicly available traffic sign dataset TT100k. The results are illustrated in Table 2. In the Table 2 and following tables, red represents the percentage decline in performance compared to the baseline, and green represents the percentage increase.

Table 2. PSG-Yolov5 ablation experiments on TT100k dataset.

Method	P/%	R/%	mAP@0.5/%	Param/M	GFLOPs
Yolov5l (baseline)	86.1	75.2	84.3	46.3	108.4
Yolov5l + GSConv	83.4 (−2.7)	76.4 (+1.2)	83.6 (−0.7)	44.5 (−1.8)	106.1
Yolov5l + GSconv + PDCM	86.2 (+0.1)	78.3 (+3.2)	86.2 (+1.9)	47.1 (+0.8)	107.3
Yolov5l + GSconv + SCAM	83.4 (−2.7)	81.2 (+6.0)	86.3 (+2.0)	45.6 (−0.7)	106.9
Yolov5l + GSconv + PDCM + SCAM	86.3 (+0.2)	82.4 (+6.2)	89.2 (+4.9)	48.4 (+2.1)	108.3

According to mAP@0.5 and Param evaluation metric in Table 2, the introduced GSConv convolution module leads to a 0.7% decrease in mAP@0.5, while our proposed PDCM structure and SCAM structure improve by 1.9% and 2.0%, respectively, and when we apply both of our proposed modules to the benchmark model the mAP@0.5 improves by 4.9%. Although the GSConv convolutional module we introduce causes a decrease in mAP@0.5, the number of parameters also decreases by 3.9%, and the inference speed is also important in the traffic sign detection task, so the decrease of slight performance is tolerable. In

summary, in terms of performance improvement alone, our proposed PSG-Yolov5 is fully validated on the TT100k dataset, and its performance improvement is obvious. After that, we tested our proposed model on Nvidia Tesla V100 and Nvidia Jetson Xavier for FPS (frames per second), respectively, the results as shown in Table 3.

Table 3. FPS detected by Nvidia Tesla V100 and Nvidia Jetson Xavier respectively.

Method	FPS (Image Size = 640 × 640)	
	Nvidia Tesla V100	Nvidia Jetson Xavier
Yolov5l	91.7	24.6
Yolov5l + GSConv	94.3	25.1
Yolov5l + GSconv + PDCM	89.3	23.5
Yolov5l + GSconv + SCAM	90.9	24.1
Yolov5l + GSconv + PDCM + SCAM	85.5	23.1

4.3.2. Comparative Experiments

In order to fully verify PSG-Yolov5 algorithm we propose in this paper, we carried out a full comparison experiment with other researchers and some advanced algorithms on TT100k dataset, and the comparison results are shown in Table 4. According to the results of comparative experiments, our proposed PSG-Yolov5 achieves relatively excellent results in both mAP@0.5 and FPS.

Table 4. Comparison with other researchers tested on TT100k.

Method	Parameters/M	GFLOPs	mAP@0.5/%	FPS
CUI [34]	-	-	77.6	-
Gan [35]	-	-	87.9	31.3
TANG [2]	-	-	93.6	2.3
CAO [18]	40.08	123.28	44.4	26
Wu [36]	-	-	79.4	41.7
Yolov3	59.58	158.00	61.7	27.0
YoloX-s	9.01	27.03	68.6	59
Mobilenet-SSD	25.067	29.20	32.0	22
Wu [37]	-	-	82.9	65
Improved-Yolov4 [38]	-	-	82.3	84.5
ReYolo [39]	-	-	68.3	188.3
ours	48.4	108.3	89.2	85.5

According to the data in Table 4, the traffic sign detection algorithm based on Cascade-RCNN of TANG [2] achieved 93.6% mAP@0.5, but its FPS was only 2.3, which could not satisfy the needs of real-time detection on embedded platforms. We also tested on the advanced Yolo series algorithm: YOLOX [40], which has lower parameters and GFLOPs, but the accuracy is not comparable to PSG-Yolov5.

In summary, our proposed algorithm perfectly achieves the balance between mAP@0.5 and FPS with slight increase in computational complexity. The inference speed is maintained while obtaining a relatively high accuracy and we tested it on an embedded platform to respond to the real-time requirements. Figure 9 shows some of the detection results of our proposed traffic detection model.



Figure 9. Some detection results on TT100k datasets.

4.3.3. Visualization of Results

To further verify the practicality of our proposed method, we refer to Grad-CAM [41] and generate the heatmaps based on the benchmark model and the PSG-Yolov5, respectively, as shown in Figure 10. Based on the information acquired by heatmaps, we can comprehend that the detection results of PSG-Yolov5 are more based on the traffic signs themselves without relying too much on the external environment, so it is less influenced by the external environment and more robust compared to the benchmark model.

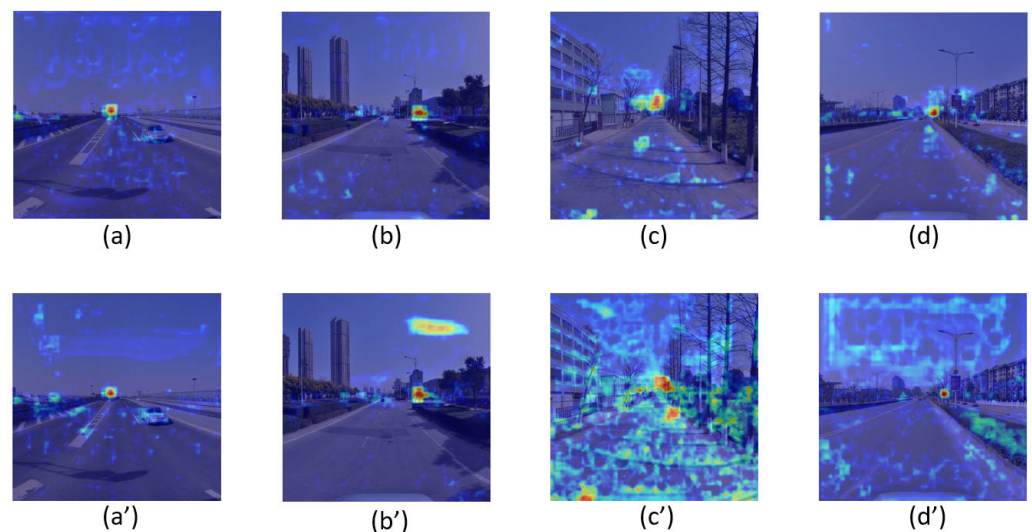


Figure 10. Heatmap visualization of selected detection results of PSG-Yolov5 and the benchmark model (Yolov5l) on the TT100k dataset. (a–d) denote the heatmap visualization of our proposed PSG-Yolov5, and (a'–d') denote the heatmap visualization based on Yolov5l.

We can conclude from Figure 10 that the detection results of our proposed PSG-Yolov5 are based more on the traffic signs themselves rather than on other external factors. In realistic roads, the task of traffic sign detection is very easily affected by weather factors. It will become unreasonable to rely too much on the external environment for traffic sign

detection under weather conditions such as rain, fog etc. This also proves from the side that our proposed algorithm has a certain robustness.

5. Conclusions

To address the problems of small target detection, information loss during multi-scale fusion, and the real-time performance of traffic sign detection algorithm, we propose the PSG-Yolov5 traffic sign detection algorithm based on Yolov5l. In this work, we propose the PDCM module, which can enhance the feature extraction ability of the model and improve the detection performance of small-scale traffic signs; our proposed SCAM module can fuse multi-scale features more efficiently and alleviate the influence of scale distribution; we also introduce the GSConv module to reduce the computational complexity of our proposed PSG-Yolov5 traffic sign detection algorithm. Through comparison experiments and ablation experiments in TT100k dataset, we can conclude that our proposed algorithm achieves significant results in terms of detection accuracy (mAP@0.5 equals 89.2%, which improves by 4.9% compared to the benchmark) and real-time detection (FPS equals 85.5). The algorithm proposed in this paper has achieved satisfying results on the TT100k dataset; however, the robustness of our algorithm in complex natural environments such as rain, snow, and fog has not yet been verified. In future, we plan to conduct research on traffic sign detection facing complex natural environments, and explore a robust TSD system.

Author Contributions: Conceptualization, J.H. methodology, Z.W. formal analysis, M.C. investigation, L.X. and N.C. supervision, W.X. writing—original draft preparation, Z.W. writing—review and editing, Z.W. visualization, Z.W. All authors have read and agreed to the published version of the manuscript.

Funding: Hubei Provincial Department of Science and Technology: 2020AAA001. The Fundamental Research Funds for the Central Universities: 2020-YB-020. The Fundamental Research Funds for the Central Universities: 2020-ZY-120.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, Z.; Liang, D.; Zhang, S.; Huang, X.; Li, B.; Hu, S. Traffic-sign detection and classification in the wild. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2110–2118.
2. Tang, Q.; Cao, G.; Jo, K.H. Integrated feature pyramid network with feature aggregation for traffic sign detection. *IEEE Access* **2021**, *9*, 117784–117794. [[CrossRef](#)]
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
4. Qi, G.; Zhang, Y.; Wang, K.; Mazur, N.; Liu, Y.; Malaviya, D. Small Object Detection Method Based on Adaptive Spatial Parallel Convolution and Fast Multi-Scale Fusion. *Remote Sens.* **2022**, *14*, 420. [[CrossRef](#)]
5. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Wang, Z. Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
6. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–27 July 2017; IEEE Computer Society: Columbia, WA, USA, 2017.
7. Li, H.; Li, J.; Wei, H.; Liu, Z.; Zhan, Z.; Ren, Q. Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles. *arXiv* **2022**, arXiv:2206.02424.
8. Quigley, M.; Conley, K.; Gerkey, B.P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. Ros: An open-source robot operating system. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
9. Takaki, M.; Fujiyoshi, H. Traffic Sign Recognition Using SIFT Features. *IEEE Trans. Electron. Inf. Syst.* **2009**, *129*, 824–831. [[CrossRef](#)]
10. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 10, pp. 886–893. [[CrossRef](#)]
11. Peng, L.; Wang, H.; Li, J. Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles. *Automot. Innov.* **2021**, *4*, 241–252. [[CrossRef](#)]
12. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]

13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)] [[PubMed](#)]
16. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
17. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
18. Cao, J.; Zhang, J.; Huang, W. Traffic sign detection and recognition using multi-scale fusion and prime sample attention. *IEEE Access* **2020**, *9*, 3579–3591. [[CrossRef](#)]
19. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [[CrossRef](#)] [[PubMed](#)]
21. Liu, Y.; Shi, G.; Li, Y.; Zhao, Z. M-YOLO: Traffic sign detection algorithm applicable to complex scenarios. *Symmetry* **2022**, *14*, 952. [[CrossRef](#)]
22. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 November 2019.
23. Wang, C.Y.; Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
24. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
25. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
26. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
27. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Boston, MA, USA, 7–12 June 2015; pp. 448–456.
28. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
29. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Beijing, China, 19–24 October 2016; pp. 516–520.
30. Rezatofighi, H.; Tsoi, N.; Gwak, J.Y.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
31. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IOU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
32. Zheng, Z.; Wang, P.; Ren, D.; Liu, W.; Ye, R.; Hu, Q.; Zuo, W. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybern.* **2021**. [[CrossRef](#)] [[PubMed](#)]
33. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
34. Cui, L.; Ma, R.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Xu, M. MDSSD: Multi-scale deconvolutional single shot detector for small objects. *arXiv* **2018**, arXiv:1805.07009. [[CrossRef](#)]
35. Gan, Z.; Wenju, L.; Wanghui, C.; Pan, S. Traffic sign recognition based on improved YOLOv4. In Proceedings of the 2021 6th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Pahang, Malaysia, 25–27 November 2021; IEEE: Piscataway, NJ, USA, 2021; Volume 6, pp. 51–54.
36. Wu, Y.; Li, Z.; Chen, Y.; Nai, K.; Yuan, J. Real-time traffic sign detection and classification towards real traffic scene. *Multimed. Tools Appl.* **2020**, *79*, 18201–18219. [[CrossRef](#)]
37. Wu, M.; Yang, J.; Zhang, W.; Zheng, Y.; Liao, J. Attention feature fusion network for small traffic sign detection. *Eng. Res. Express* **2022**, *4*, 035047. [[CrossRef](#)]
38. Wu, X.; Cao, H. Traffic Sign Detection Algorithm Based On Improved YOLOv4. *J. Phys. Conf. Series.* **2022**, *2258*, 012009. [[CrossRef](#)]
39. Zhang, J.; Zheng, Z.; Xie, X.; Gui, Y.; Kim, G.J. ReYOLO: A Traffic Sign Detector Based on Network Reparameterization and Features Adaptive Weighting. *J. Ambient. Intell. Smart Environ.* **2022**, *14*, 317–334. [[CrossRef](#)]

-
40. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
 41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.