Neural Computing and Applications (2022) 34:2233–2245
https://doi.org/10.1007/s00521-021-06526-1

ORIGINAL ARTICLE

A real-time and high-precision method for small traffic-signs recognition

Junzhou Chen[1] · Kunkun Jia[1] · Wenquan Chen[1] · Zhihan Lv[2] · Ronghui Zhang[1]

Received: 12 March 2021 / Accepted: 8 September 2021 / Published online: 25 September 2021

Abstract
As a fundamental element of the traffic system, traffic signs reduce the risk of accidents by providing essential information about the road condition to drivers, pedestrians, etc. With the rapid progress of computer vision and artificial intelligence, traffic-signs recognition systems have been applied for the advanced driver assistance system and auto driving system, to help drivers and self-driving vehicles capture the important road information precisely. However, in real applications, small traffic-signs recognition is still challenging. In this article, we propose an efficient method for small-size traffic-signs recognition, named traffic-signs recognition small-aware, with the inspiration of the state-of-the-art object detection framework YOLOv4 and YOLOv5. In general, there are four contributions in our work: (1) for the Backbone of the model, we introduce high-level features to construct a better detector head; (2) for the Neck of the model, receptive field block-cross is utilized for capturing the contextual information of feature map; (3) for the Head of the model, we refine the detector head grid to achieve more accurate detection of small traffic signs; (4) for the input, we propose a data augmentation method named Random Erasing-Attention, which can increase difficult samples and enhance the robustness of the model. Real experiments on the challenging dataset TT100K demonstrate that our method can achieve significant performance improvement compared with the state of the art. Moreover, it is a real-time method and shows huge potential applications in advanced driver assistance system and auto driving system.

Keywords Traffic-signs recognition · Convolutional neural network · Object detection · Data augmentation

1 Introduction

As an essential part of the advanced driver assistance system(ADAS) and auto driving system(ADS), traffic-signs recognition(TSR) technology can help drivers and self-driving vehicles capture the important road information, as illustrated in Fig. 1. TSR is a technology by which a vehicle can recognize the traffic signs put on the road (e.g., "speed limit" or "children"). A TSR system generally consists of two phases of detection and classification. The detection generally uses the shape and color characteristics of traffic signs to extract traffic signs from natural scenes. Classification is to identify the content of the detected traffic signs. Traffic-sign recognition is of great significance for road accident avoidance [1].

In general, there are two eras of traffic-signs recognition methods, one is the traditional method (hand-crafted features), and the other is the deep learning method (learning features with deep neural networks). In traditional methods, the pipeline generally includes extracting regions of interest (including traffic signs), extracting features, and then sending these features to the classifier (e.g., SVM) [3–8]. However, traditional methods have a poor performance in multi-class tasks. In recent years, deep learning methods have shown superior performance for many tasks such as image classification and detection. For object detection, deep learning methods (e.g., YOLO [9], Faster R-CNN [10]) perform well on mainstream benchmarks

✉ Ronghui Zhang
zhangrh25@mail.sysu.edu.cn

[1] Guangdong Provincial Key Laboratory of Intelligent Transport System, School of Intelligent Systems Engineering, Shenzhen Campus of Sun Yat-sen University, No. 66, Gongchang Road, Guangming District, Shenzhen, Guangdong 518107, P.R. China

[2] School of Data Science and Software Engineering, Qingdao University, Qingdao 266071, China

Fig. 1 Traffic-signs recognition(TSR) system helps drivers and self-driving vehicles capture the important road information [2]

(e.g., Pascal VOC [11] and MS COCO [12]) in terms of accuracy and speed.

Traffic-sign recognition is a subtask of object detection, just like face recognition. Several benchmarks widely used to evaluate TSR performance are German Traffic Sign Benchmark(GTSB) [13, 14], Laboratory for Intelligent and Safe Automobiles (LISA) [15] and Tsinghua-Tencent 100K (TT100K) [16]. In traffic sign detection tasks, a sign might only occupy 30×30 pixels, in a high-resolution image of 2048×2048 pixels. A typical example in Tsinghua-Tencent 100K benchmark [45] is visualized in Fig. 2.

However, small object detection is still challenging for object detection due to its low resolution and less information. In recent years, several approaches have been devoted to improving the performance of small object detection. One of these methods [17–19] is to build a high-resolution feature map and make predictions on it. This method obtains fine detail information and compensates for the lack of information, but also loses contextual information. Naturally, the results are not satisfactory. Another effective method [20–24] improves the performance of small object detection by building a top-down structure with skip connections. This system combines low-level details and high-level semantic features at all scales, which can effectively improve detection accuracy. However, due to the increasing complexity of the network, these methods suffer from high computational costs during the training and testing phases and cannot maintain real-time detection. Moreover, these methods usually obtain small feature maps by downsampling several times and then reconstruct the spatial resolution. In fact, small feature maps retain little information for small object detection, and once the signal is lost due to downsampling, it is almost impossible to recover. Therefore, for small object detection, it is necessary for neural networks to increase the receptive field of neurons while maintaining the spatial structure.

Inspired by the above methods, we propose TSR-Small-Aware(TSR-SA), to solve small object detection problems in this article, as illustrated in Fig. 2. For the model, we choose YOLOv4 [25], the object detector with the strongest overall performance, as the baseline of the model. For the benchmark, we choose TT100K [16], which contains a large number of small-size traffic signs in the natural scene, to evaluate the model. Although YOLOv4 has a strong performance in general object detection($mAP@0.5 = 65.7\%$ in MS COCO [12]). Its performance can be improved in specific tasks, such as small-size traffic-signs recognition. After multiple downsampling, although the information of the feature map can support the detection of

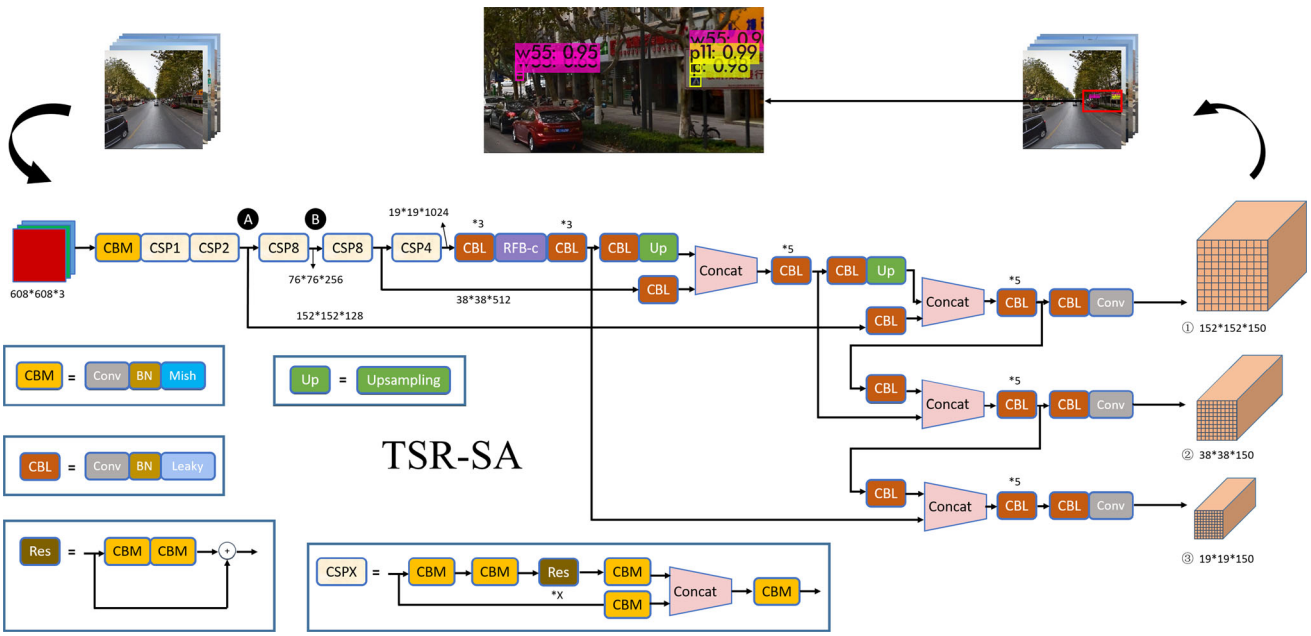

Fig. 2 The network structure of TSR-SA

medium and large-sized objects, it cannot meet the detection of small-sized objects. Therefore, we made some changes to YOLOv4. First, for the Backbone of the model, we introduce high-level features to construct a better detector head. Second, for the Neck of the model, receptive field block-cross(RFB-c) is utilized for capturing the contextual information of the feature map. Third, for the Head of the model, we refine the detector head grid to achieve more accurate detection of small traffic signs. Ultimately, for the input, we propose a data augmentation method named Random Erasing-Attention(RE-A), which can increase difficult samples and enhance the robustness of the model.

To summarize, the contributions of this article are as follows:

(1) We propose a novel framework TSR-Small-Aware(TSR-SA) and a data augmentation method Random Erasing-Attention(RE-A) for small object detection.
(2) TSR-SA is devised to improve small object detection. As an important component of TSR-SA, RFB-c separates the most significant contextual features and improves the performance of small object detection without additional computational cost. The neck and head are designed to aggregate low-level details and high-level semantic features.
(3) RE-A data augmentation is proposed to increase the number of difficult samples and improve the robustness of the model. RE-A generates some occluded traffic signs, which makes the model perform better for occlusion problems.
(4) Real experiments on the challenging dataset TT100K demonstrate that TSR-SA has a significant improvement compared with the state of the art, while still running at real-time speed.

The rest of the paper is organized as follows: In Sect. 2, we discuss the related work on object detection, traffic-signs recognition and small object detection. Details of our network are given in Sect. 3. Experimental results and implementation details are presented in Sect. 4. Discussions and conclusions are introduced in Sect. 5.

2 Related work

In recent years, deep convolutional neural networks or deep learning methods have gradually replaced traditional methods in various tasks and become mainstream methods. For object detection, the performance of deep learning methods is at a high level on mainstream benchmarks (e.g., Pascal VOC [11], MS COCO [12]) and still getting better. And many researchers begin to apply these methods to traffic-signs recognition. In this article, we focus on small traffic-signs recognition, and then, we discuss related research work on object detection, traffic signs detection, and small object detection.

2.1 Object detection

CNN-based object detection methods can be grouped into two genres [26]: "one-stage" and "two-stage". The two-stage methods first extract region proposals and then classify and regress each proposal to achieve detection results. The mainstream two-stage methods include R-CNN [27], SPPNet [28], Fast R-CNN [29], Faster R-CNN [10], etc. But the two-stage approaches incur a lot of computational costs. One-stage methods discard the stage of generating region proposals, in order to accelerate the inference speed and achieve real-time detection. The representative of one-stage methods includes YOLO [9], SSD [30], and RetinaNet [31] (Figs. 3, 4).

Two-stage detection Two-stage methods start from R-CNN [27], which uses the CNN network to extract image features, from experience-driven artificial features (e.g., HOG [32], SIFT [33]) to data-driven feature learning. SPPNet [28] proposes a spatial pyramid pooling (SPP) layer, which can generate fixed-size feature vectors of ROI without resizing. Fast R-CNN [29] uses softmax instead of SVM in R-CNN for classification. And soon Faster R-CNN [10] proposes RPN network, truly achieving end-to-end training.

One-stage detection The two-stage methods are slightly deficient in real-time performance. To address this problem, many one-stage methods have been proposed. YOLO [9](You Only Look Once) is the first one-stage method, which divides each image into a $S \times S$ grid, and then, each grid is responsible for detecting those objects whose center point falls within the grid. The author has made a series of improvements based on YOLO and has already proposed YOLOv2 [34], v3 [35], v4 [25]. SSD

Fig. 3 Tsinghua-Tencent 100k

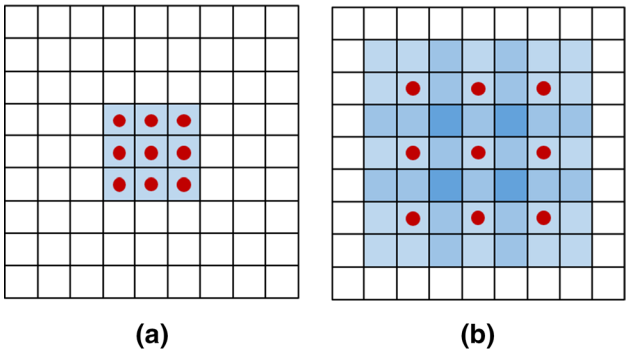

Fig. 4 Dilated convolution. The left: dilatation *rate* = 1, receptive field = 3 × 3; The right: *rate* = 2, receptive field = 7 × 7

[30] is the second one-stage method, and its main contribution is to propose multi-scale features for object detection. It significantly improves the accuracy of the one-stage method, especially for small objects. RetinaNet [31] proposes a new loss function named focal loss to solve the extreme foreground-background class imbalance encountered during the training of dense detectors.

2.2 Traffic-signs recognition

Traditional traffic-signs recognition methods can be divided into two categories: color-based and shape-based, as mentioned in [36]. Now, it has become a consensus that machine learning-based methods are superior to traditional methods. Among the machine learning-based methods, the deep learning-based or CNN-based methods have the best performance and have become the mainstream method of traffic-signs recognition.

Traditional methods Although the appearance of traffic in each country is different, it has the same characteristics in terms of color and shape. For example, red usually indicates a prohibition sign. Traditional methods utilize these characteristics to design detectors, such as [37–39](color-based), [40, 41](shape-based methods).

Machine learning-based methods Machine learning-based methods can be divided into three stages: AdaBoost-based methods [14, 42, 43], SVM-based methods [5, 44, 45], CNN-based methods [9, 30, 31]. With the development of deep learning, CNN-based methods have gradually become the mainstream of traffic-signs recognition. Traffic-signs recognition is essentially an object detection problem. Therefore, many researchers apply advanced object detection methods to traffic signs detection. In this article, we focus on the application of YOLOv4 [25] in traffic-signs recognition.

2.3 Small object detection

Small object detection is a challenging task of object detection. Compared with medium and large objects, small objects are more difficult to find and locate. First, small objects cover fewer pixels, which means that the features for detection are insufficient and the feature representation is weak; secondly, due to their small size, the object may appear anywhere in the input image, such as corners or areas that overlap with other objects. More information is the key to small object detection. The current strategies can be divided into four categories [46]: Multiscale Representation [24, 47, 48], Contextual Information [49, 50], Super-Resolution [51, 52], Region Proposal [53, 54]. These strategies can help CNN to extract richer features when detecting small objects, thereby improving the accuracy of small object detection.

For Contextual Information, there are several methods [50, 55, 56], which use **dilated convolution** to enrich semantic information of feature map. The initial proposal of dilated convolution is applied to image segmentation [55]. Segmentation algorithms typically use pooling and convolution layer to increase the receptive field, but also narrowed the resolution. To address this issue, dilated convolution was proposed to increase the receptive field while maintaining resolution. In computer vision, this is actually a universal method. Later dilated convolution was applied to object detection, such as receptive field block(RFB) [50], which is a convolution block used to capture multi-scale contextual information in feature maps.

3 Method

In this section, we first briefly introduce the baseline YOLOv4 and its previous versions [9, 25, 34, 35, 57]. Secondly, we introduce receptive field block-cross (RFB-c), which is designed to fuse low-level detailed and high-level semantic features. Then describe the details of the proposed TSR-SA and finally introduce the data augmentation method Random Erasing-Attention(RE-A).

3.1 Preliminaries

YOLO [35] In the YOLO detection pipeline, all bounding boxes and class probabilities are output by a network, which realizes end-to-end detection. Yolo's CNN network divides the input image into $S \times S$ grids. Then, each grid is responsible for detecting the target whose center point falls within the grid. Each grid will predict B bounding boxes, the confidence score p of the bounding box, and the category C. The size and position of the

bounding box can be characterized by four values: (x, y, w, h), where (x, y) is the center coordinate of the bounding box, and w and h are the width and height of the bounding box. To summarize, each cell needs to predict $B \times (5 + C)$ values. If the input image is divided into $S \times S$ grids, then the final predicted value is a tensor of the size of $S \times S \times (B \times (5 + C))$, usually $B = 3$ in YOLO.

YOLOv4 [25] Bochkovskiy et al. proposed the YOLOv4 in April 2020. Compared with YOLOv3, YOLOv4 integrates many of the most advanced methods today, such as Mosaic data augmentation, Cross-stage partial connections (CSP) [58], Mish activation [59], SPP-block [28], PAN path-aggregation block [60]. YOLOv4 verifies the influence of these state-of-the-art methods of object detection during the detector training. It is an efficient and powerful object detection model. It makes everyone can use a 1080 Ti or 2080 Ti GPU to train a super fast and accurate object detector.

3.2 Receptive field block-cross

The pipeline of receptive field block-cross(RFB-c) is devised to build the most significant contextual features. RFB-c is a multi-branch convolutional block, as illustrated in Fig. 5. Its inner structure can be divided into four branches. Each branch is composed of three components: CBL, CBL-d, and the shortcut connection from the input layer. Eventually, the feature tensors from the four branches are concatenated into a multi-scale contextual feature tensor to the output layer.

Specifically, firstly, we utilize the bottleneck [61] structure in each branch, consisting of a 1×1 CBL, to reduce the number of channels (e.g., from 512 to 256) in the feature map. Second, n stacked CBL-d(kernel size is $k \times k (k > 1)$), are employed to capture the contextual information without increasing the parameters. Third, we use an 1×1 CBL to increase the number of channels from 256 to 512. Then, a shortcut connection [62] with the input layer is used on each main branch. Ultimately, the four feature tensors ($19 \times 19 \times 512$) are concatenated into a multi-scale contextual feature tensor ($19 \times 19 \times 2048$).

- *CBL and CBL-d* A CBL block is composed of a convolutional layer, batch normalization, and leaky activation function. The components of a CBL-d block are the dilated convolutional layer, batch normalization, and leaky activation function. The only difference from CBL is using **dilated convolution**.
- *Dilated convolution* The basic intention of dilated convolution is to generate a multi-scale feature map, capturing contextual information without increasing the amount of parameters. As illustrated in Fig. 4, the left is a dilated convolution kernel with dilatation $rate = 1$ and the right is $rate = 2$. Only the element at the red dot is operated, and the remaining elements are skipped. The blue area represents the size of the receptive field, with 3×3 on the left and 7×7 on the right. The formula for the equivalent convolution kernel size and receptive field of the dilated convolution is as follows [49, 55, 63]:

$$\hat{k} = k + (k - 1)(r - 1) \tag{1}$$

where k is set to 3, $\hat{k}$ is equivalent kernel size after dilation, r is the dilation rate,

$$RF_{n+1} = RF_n + (\hat{k} - 1) \times \prod_{i=1}^{n-1} s_i \tag{2}$$

where RF_n is the receptive field of layer $n (n > 1)$, s_i is the stride of layer i, s_i is set to 1. In particular, when $n = 1, k = 3, s_i = 1, RF_0 = 3$.

3.3 TSR-SA

RFB-c is flexible enough to plug into any backbone network and off-the-shelf detectors. Considering the trade-off between accuracy and efficiency, we assembled RFB-c into the one-stage framework YOLOv4 for demonstration. YOLOv4 has a poor performance in small object detection. Because after multiple downsampling, the feature map has little spatial information for small instances.

In this section, we propose TSR-small-aware (TSR-SA) as shown in Fig. 2. The proposed method pipeline consists of four stages: (1) Backbone, for extracting base features; (2) RFB-c, for building contextual features; (3) Neck, for fusing low-level and high-level features; (4) Head, for making predictions. We will describe the three stages in detail below.

3.3.1 Backbone

In this stage, the image is resized to 608×608 as input and then used in the convolutional neural network for feature extraction, similar to the traditional architecture. For this part, various backbone networks can be applied, such as VGG and ResNet. In this article, CSPDarknet53 [58] is selected as the backbone network.

3.3.2 RFB-c

In the previous Sect. 3.2, we described the structure of RFB-c in detail. We devise the RFB-c block and add it over the backbone, since it significantly increases the receptive field, builds the most significant context features, and

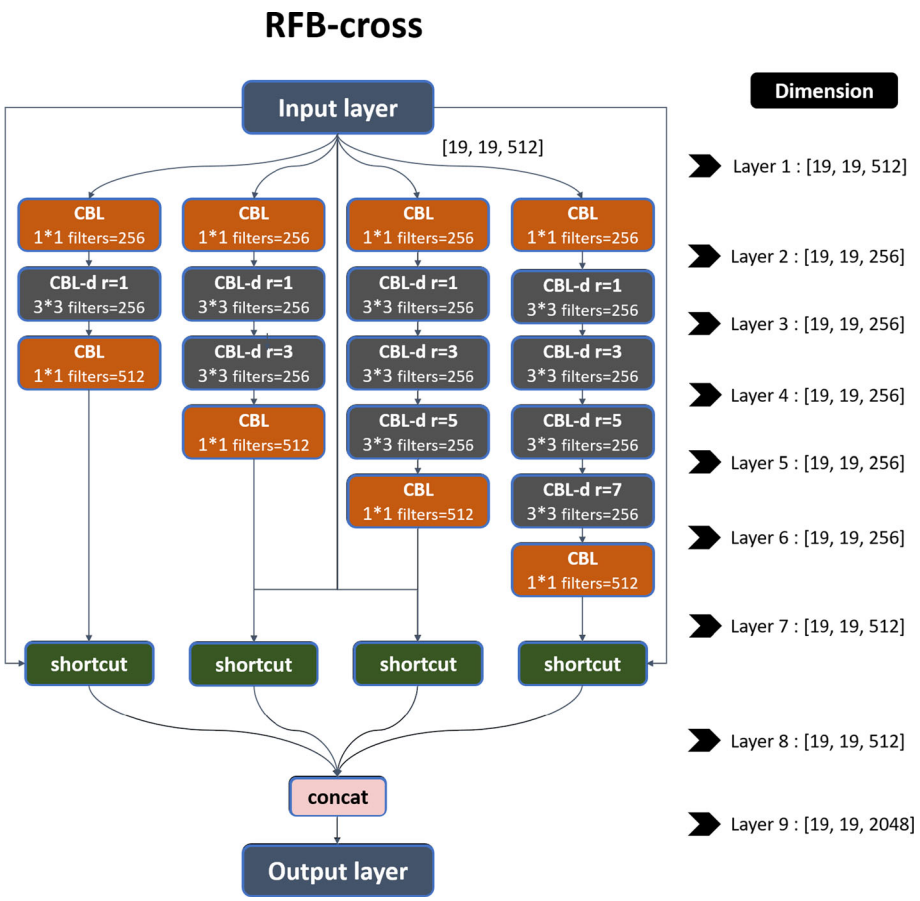

Fig. 5 Receptive field block-cross. RFB-c has four branches, which are composed of CBL and CBL-d

causes almost no reduction of the network operation speed. The input of RFB-c is a feature with dimension $19 \times 19 \times 512$. Four features are obtained after four branches of different scales. Finally, these four features are concatenated into one feature with a dimension of $19 \times 19 \times 2048$, which separates the context information.

3.3.3 Neck

In this stage, we build the top-down structure to fuse low-level and high-level features from different backbone levels for different detector levels. And PANet is selected as the neck. We introduce the low-level features of the network backbone into the neck to solve the problem of insufficient feature map information. As illustrated in Fig. 6, the dotted line at B is the lower-level feature, and its dimension is $76 \times 76 \times 256$, and the solid line at A is the higher-level feature, and its dimension is $152 \times 152 \times 128$. These low-level features are not only used for the detector head-1 but also the other two detector heads-2,3. So this can not only improve the detection accuracy of small objects but also relatively improve the detection accuracy of medium and large targets. (Note that the three detector

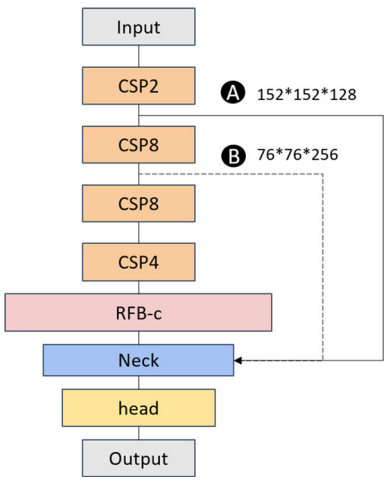

Fig. 6 Introduce low-level features (152×152) instead of high-level features (76×76)

heads of YOLO are more effective in detecting small, medium, and large objects, respectively.)

3.3.4 Head

Following YOLOv4, we apply three scales to detect objects on the feature map output by neck. We refined the

detection grid of the detector head in order to accurately locate small traffic signs. We use 152×152 grids to divide the picture instead of 76×76 in the detector head-1, as illustrated in Fig. 7 Each grid is responsible for predicting objects whose center point falls within its range. Therefore, the 152×152 grids is not easy to miss small objects, thereby improving the ability to locate small objects. And 150 in the tensor $(152 \times 152 \times 150)$ represents $(3 \times (5 + 45))$, as mentioned in Sect. 3.1, where $B = 3$ (The anchor box has 3 sizes), $C = 45$(There are 45 classes of traffic signs, as shown in Fig. 3), 5 is the dimension of (p, x, y, w, h).

3.4 Random erasing-attention

Random Erasing-Attention (RE-A) is devised to generate hard samples. It is more effective for the detection of occluded traffic signs. The difference between RE-A and Random Erasing(RE) [64] is that the former focuses on the traffic signs, while the latter focuses on the entire picture, as illustrated in Fig. 8. We apply the attention mechanism in RE. The original object detection is not only for traffic signs but for general object detection (e.g., person, car, cat, flower). These objects appear in any corner of the picture, so pay attention to the whole picture. But now we only detect traffic signs, so we need to focus on the part of the traffic signs, not the entire picture. The following is our algorithm:

Algorithm 1: Random Erasing-Attention(RE-A)

Input: An image X;
(x, y, w, h) of each labeled object;
Maximum erasing probability P;
Area ratio Ar_l and Ar_h;
Aspect ratio r_l and r_h
Output: A new image X^*

Initialization: $p \leftarrow Rand(0, 1)$
if $p \geq P$ **then**
  $X^* \leftarrow X$
else
  Get the (x, y, w, h) of the labeled object;
  Get the erasing value $\gamma \leftarrow Rand(0, 255)$;
  $S = h \times w$;
  $Ar \leftarrow Rand(Ar_l, Ar_h)$;
  $S_e = S \times Ar$;
  $r_e \leftarrow Rand(r_l, r_h)$;
  $H_e = S_e \times r_e$, $W_e = S_e \times r_e$;
  $\mathcal{M} = (H_e, W_e, h, w)$;
  $X^* \leftarrow Mask(X, \mathcal{M})$
end
return X^*

Inspired by [64], for an image, RE-A randomly selects some pixels in a rectangular area of traffic signs and replaces them with random values γ. r_e is aspect ratio, defined as $r_e = H_e/W_e$ between minimum r_l and r_h, where H_e and W_e is the height and width of the region. Therefore, the area of the erasing region is denoted as $S_e = H_e \times W_e$, and the size is represented as $H_e = \sqrt{S_e \times r_e}$ and $W_e = \sqrt{S_e / r_e}$. Furthermore, assuming that the size of the features is $S = H \times W$, we define the area ratio Ar as $Ar = S_e/S$. Therefore, S_e can be obtained by a random area ratio Ar, which is initialized between minimum Ar_l and maximum Ar_h. Formally, we formulated the area of the erasing region S_e as $S_e = S \times Ar$.

Specifically, the RE-A mask is binary in the same shape of the selected region. And the mask is filled with 0 when the region is selected to be erased. Otherwise, all is set to 1. Therefore, RE-A can be formulated as:

$$X^* = Mask(X, \mathcal{M}) = \mathcal{M} \cdot X + (1 - \mathcal{M}) \cdot X \tag{3}$$

where $\mathcal{M}$ denotes the RE-A mask. Pseudocode of RE-A applied to an image is detailed in Algorithm 1.

4 Experiments

4.1 Dataset

The data in TT100K [16] comes from Tencent Street View Map, as illustrated in Fig. 3. 100,000 cropped images are obtained after preprocessing. The resolution of the image is 2048×2048. Among them, 10,000 annotated pictures contain 30,000 traffic signs. TT100K is a challenging data set, which contains many **small-size** traffic signs, covering changes in weather and illumination (Fig. 9).

4.2 Training details

The TT100K dataset includes approximately 150 classes of traffic signs. In the training, we ignored those categories with instances less than 100 following Zhu et al. [16]. Finally, there are 45 classes of traffic signs for research. The benchmark dataset is publicly available at http://cg.cs.tsinghua.edu.cn/traffic-sign/. In [16], the classes containing 100 to 1000 instances are augmented to 1000 instances by pasting traffic signs into new street view images. Other classes with more than 1000 instances remain unchanged. To avoid the impact of different augmented policies, TSR-SA and other methods are all experimented on the original training set and test set. The training set has 6105 images with a resolution of 2048×2048, which contains about 15,000 traffic signs belonging to 45 categories. While there are about 3071 pictures and 7700 traffic signs in the test set. In addition, in training, we used RE-A to generate about 2000 hard samples to replace the samples corresponding to the training set.

The experiment environment is based on the Darknet library. The experiment equipment is the machine with NVIDIA Tesla V100 GPUs. We pre-train YOLOv4 on the MS COCO dataset to get the pre-training parameter weight (YOLOv4.conv.137), and then, we fine-tune our networks TSR-SA on TT100K. For hyperparameters, we set the initial learning rate to 0.001, which is reduced to one-tenth of the original at 72k and 81k iterations. The total number of iterations is 90k, and the batch size is set to 16 (Fig. 10).

4.3 Result

As illustrated in Table 1, the mAP of our method is **90.2%** on the TT100K dataset, which is state of the art. We also trained the latest YOLOv4 (published in April 2020) and YOLOv5 (published in June 2020) on TT100K. YOLOv5-x is the largest model in the YOLOv5 series, followed by YOLOv5-l. The mAP of YOLOv4 is 88.4% and that of YOLOv5x is 88.1%. The mAP of the two is very close. The mAP of YOLOv5-l is 86.8%, which is lower than the previous two. MSA_YOLOv3 published in March 2020 is based on YOLOv3, and its mAP is 86.3%. CAB was published in June 2020, and its mAP is 78.0%. Since Faster R-CNN was published in 2016, its mAP is 52.9%. Besides, we also compared the mAP of each category in the TT100K dataset of mainstream methods, as illustrated in Table. 3. In general, our method (TSR-SA) has a significant improvement compared to the latest research results.

Although we did a very good job on this task, there is still a problem. Our method overall performance (mAP) is the best, but there are anomalies in a few categories. It can be observed that YOLOv4 and YOLOv5-x are our strong opponents. As illustrated in Table 3, the performance of TSR-SA, YOLOv4, and YOLOv5-x will not exceed 5% (AP) in most categories. But in a few categories, their performance differed by 10%(AP) or even more. For example, for il100, the AP of our method is 99.9%, while YOLOv4 is 89.1%; for ph4.5, ours is 91.8%, and YOLOv5-x is 77.6%. Not only that, for p6, YOLOv4's AP is 86.0%, while ours is only 68.5%; for w32, YOLOv5-x is 85.6%, while ours is 71.2%, and YOLOv4 is even only 59.5%. After analysis, we believe that this is due to **the imbalance of categories**. As shown in Fig. 9, some categories have more than 2500 instances, and some have only 100 instances. The number of instances in the above categories (il100, ph5, p6, w32) does not exceed 150. Neural networks may tend to categories with more instances. To solve this problem, the author of TT100K, augmented the classes

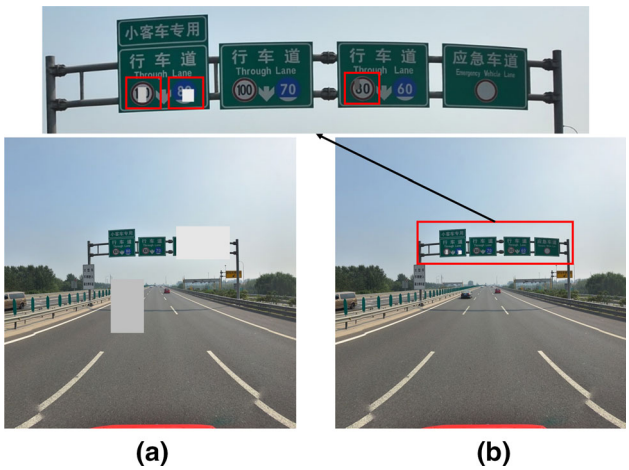

Fig. 8 The left: Random Erasing (RE); The right: Random Erasing-Attention (RE-A)

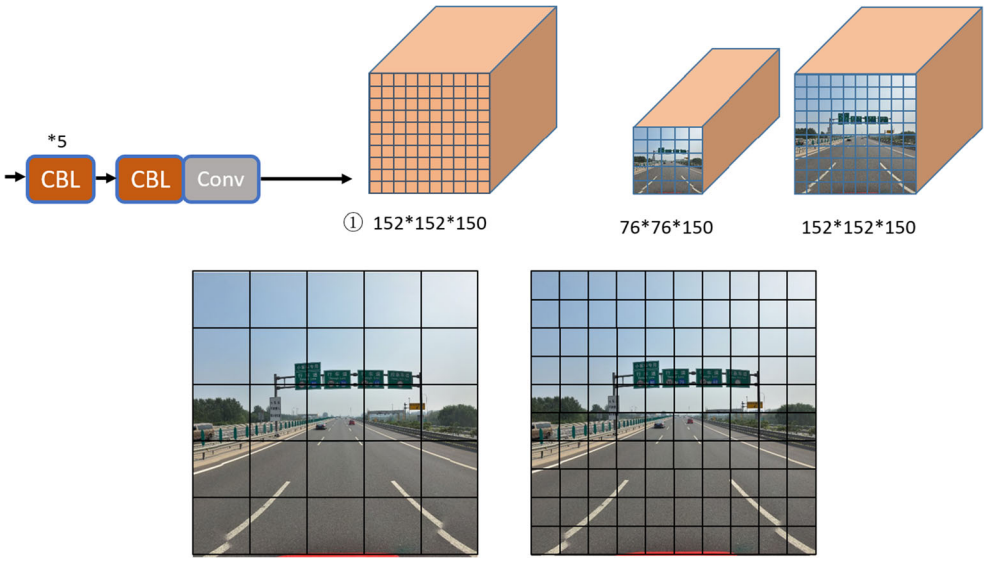

Fig. 7 The mesh of the detector head-1 is refined to 152 × 152

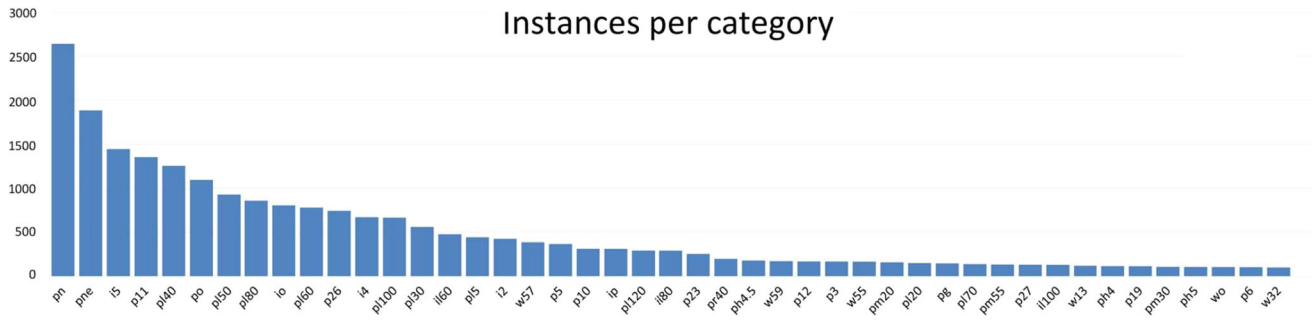

Fig. 9 Number of instances in each class, for classes with more than 100 instances [16]

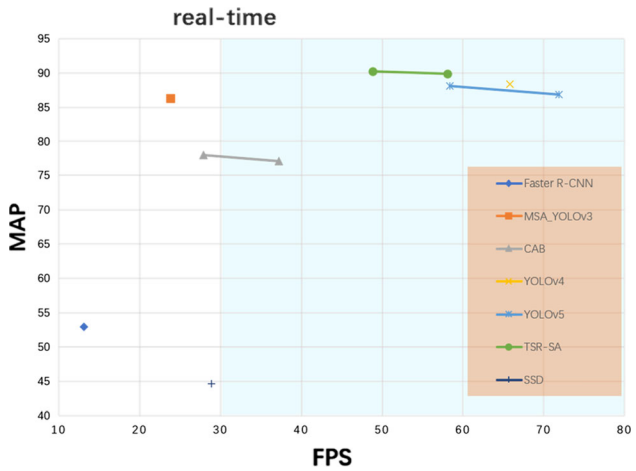

Fig. 10 Comparison of the speed and accuracy of different object detectors. Blue area: Real-time detection (> 30 FPS)

with between 100 and 1000 instances in the training set to give them 1000 instances by pasting traffic signs into new street view images. To avoid the impact of different augmented policies, following [49], we do not currently use artificial synthesis to increase the amount of data.

4.4 Inference speed

1950 images in the TT100K dataset are selected to evaluate the speed of our model. The batch size is set to 1. Inference speed faster than 30FPS is considered to be able to achieve real-time detection as shown in Fig. 11. As illustrated in Table 1, when the image input size is 608×608, the inference speed of our model can reach FPS 58.1 (0.0172s, without RFB-c) and FPS 48.8 (0.0205s, with RFB-c). Under the same conditions, the inference speed of YOLOv4 can reach FPS 65.8 (0.0152s), which is slightly faster than ours. And inference speed of YOLOv5-x can reach FPS 58.4 (0.0171s), and YOLOv5-l can reach FPS 71.9 (0.0139s), which is the fastest. But our method maintains **high precision** while performing **real-time detection**, which is of great significance for ADS and ADAS.

4.5 Ablation studies

We investigate the effectiveness of different components of TSR-SA. All experiments are performed on the TT100K dataset. As shown in Table 2, "RE-A" refers to the use of Random Erasing-Attention for data augmentation of the image, "RFB-c" means the use of receptive field block-

Table 1 Comparison of speed and accuracy of each method on TT100K

Method	mAP(%)	Resolution	Speed (s)	FPS	GPU
Faster R-CNN [10]	52.9	–	0.0760	13.1	–
SSD [19]	44.6	512×512	0.0346	28.9	1080 Ti
MSA_YOLOv3 [65]	86.3	544×544	0.0420	23.8	Tesla P100
CAB [49]	78.0	512×512	0.0360	27.9	1080 Ti
CAB-s [49]	77.1	512×512	0.0269	37.2	1080 Ti
YOLOv4 [25]	88.4	608×608	0.0152	65.8	V100
YOLOv5-l [57]	86.8	608×608	**0.0139**	**71.9**	V100
YOLOv5-x [57]	88.1	608×608	0.0171	58.4	V100
Ours(without RFB-c)	**89.9**	608×608	0.0172	58.1	V100
Ours(with RFB-c)	**90.2**	608×608	0.0205	48.8	V100

Fig. 11 Detection examples on the TT100K testing set. The target area is enlarged and displayed. Zoom in to see more details

cross, "Neck & Head" represents the modified Neck and Head, more details are described in Sect. 3.

Table 2 presents that When RE-A, RFB-c, Neck & Head are applied separately, mAP increases by 0.5%, 0.6%, and 0.9%, respectively. RE-A belongs to image preprocessing and will not change the model structure, so it will not affect the inference speed. While RE-c and Neck & Head slow down the inference speed to a certain extent, the model is still far higher than the real-time detection standard. When they are used in combination, combination 1 (RE-A, RFB-c) reaches mAP 89.6%, combination 2 (RE-A, Neck & Head) 89.9%, combination 3 (RFB-c, Neck & Head) 89.8%. When the three are used together, mAP reaches 90.2%, which is the current state of the art for real-time detection. We found that when RE-A, RFB-c, and Neck & Head are used in combination, the performance of the model can still be improved, which shows that the roles of these three modules do not conflict (Table 3).

Table 2 Ablation studies of components (TSR-SA, TT100K)

RE-A	RFB-c	Neck & Head	mAP (%)	FPS
			88.4	65.8
✓			88.9	65.8
	✓		89.0	55.9
		✓	89.3	58.1
✓	✓		89.6	55.7
✓		✓	89.9	58.1
	✓	✓	89.8	48.9
✓	✓	✓	90.2	48.8

5 Conclusions and discussions

Our effort in this paper is to improve the speed and accuracy of small traffic-signs recognition. Traffic-signs recognition is a subtask of object detection, and an essential part of ADAS and ADS. Small object detection has always been a challenge for object detection. Although the previous method has achieved good results on this subject, its accuracy and speed are still not up to the ideal level.

This article proposed an efficient framework TSR-SA and data augmentation method RE-A for small traffic-signs recognition, which is a challenging subject. The RFB-c block of TSR-SA separates out the most significant contextual features and improves the performance of small object detection without additional computational cost. Innovative neck and head effectively aggregate low-level details and high-level semantic features. Additionally, we propose a data augmentation method named RE-A, which can increase difficult samples and improve the performance of detecting occluded objects. Real experiments show that TSR-SA has reached the state of the art on TT100K, with a real-time speed. For ADAS and ADS, it is an advancement of the traffic-signs recognition system; for computer vision, it is a contribution of object detection.

Currently, mainstream TSR methods and public datasets mainly involve daytime and normal weather scenarios. Rare methods and datasets focus on traffic-sign recognition at night and in extreme weather. There are many difficulties in recognizing traffic signs at night, such as headlight reflection, taillight interference, and streetlight illumination. Extreme weather, such as fog, heavy rain, and blizzards, can seriously affect the quality of images captured by the camera. These extreme situations require new datasets and methods to address, and nighttime scenarios and weather conditions are future research directions. In the future, we will focus on night-time traffic-sign

Table 3 Comparison of mAP of each class on TT100K

Method	Total	i2	i4	i5	il100	il60	il80	io	ip
CAB	78.0	76.0	81.5	89.4	80.6	89.9	85.3	80.5	78.0
MSA_YOLOv3	86.3	85.0	84.0	92.0	85.0	95.0	89.0	85.0	90.0
YOLOv4	88.4	82.4	93.7	96.5	89.1	99.8	**98.6**	84.5	82.5
YOLOv5-x	88.1	88.6	**94.7**	**97.1**	94.3	98.6	94.9	**90.6**	**95.8**
Ours	**90.2**	**88.9**	93.1	96.6	**99.9**	**99.9**	97.0	85.6	87.7

Method	Total	pl0	p11	p12	p19	p23	p26	p27	p3
CAB	78.0	69.1	77.6	74.3	87.6	87.1	81.4	81.0	74.7
MSA_YOLOv3	86.3	74.0	72.0	78.0	73.0	82.0	81.0	83.0	81.0
YOLOv4	88.4	**85.3**	79.6	**94.5**	**89.3**	**95.6**	90.1	95.8	89.3
YOLOv5-x	88.1	81.9	87.9	83.9	82.7	92.7	92.1	94.3	93.2
Ours	**90.2**	84.6	**90.5**	90.1	81.1	92.6	**93.5**	**99.1**	**95.9**

Method	Total	p5	p6	pg	ph4	ph4.5	ph5	pl100
CAB	78.0	84.5	82.5	87.5	71.8	64.4	79.2	88.4
MSA_YOLOv3	86.3	77.0	72.0	92.0	82.0	83.0	63.0	88.0
YOLOv4	88.4	96.7	**86.0**	86.5	**85.3**	86.3	86.0	94.6
YOLOv5-x	88.1	91.3	80.8	**96.1**	80.6	**92.7**	77.6	**95.6**
Ours	**90.2**	**97.4**	68.5	83.7	84.2	88.5	**91.8**	93.0

Method	Total	pl120	pl20	pl30	pl40	pl5	pl50	pl60
CAB	78.0	87.9	68.6	73.3	74.8	19.3	75.1	76.3
MSA_YOLOv3	86.3	80.0	75.0	74.0	74.0	78.0	72.0	76.0
YOLOv4	88.4	91.5	**86.3**	86.0	88.2	92.3	83.8	89.2
YOLOv5-x	88.1	93.2	80.2	87.7	88.6	89.0	86.7	88.2
Ours	**90.2**	**96.1**	81.9	**88.1**	**93.3**	**94.0**	**87.0**	**89.9**

Method	Total	pl70	pl80	pm20	pm30	pm55	pn
CAB	78.0	72.9	78.8	73.8	67.3	80.5	85.4
MSA_YOLOv3	86.3	79.0	72.0	76.0	67.0	71.0	83.0
YOLOv4	88.4	87.2	90.6	**93.2**	78.9	**94.7**	85.9
YOLOv5-x	88.1	80.2	91.5	89.5	**82.8**	**94.7**	91.6
Ours	**90.2**	**90.9**	**92.9**	89.8	80.4	94.5	**94.3**

Method	Total	pne	po	pr40	w13	w32	w55
CAB	78.0	89.5	63.5	88.9	70.7	66.8	83.5
MSA_YOLOv3	86.3	96.0	76.0	85.0	70.0	91.0	79.0
YOLOv4	88.4	95.6	78.4	**99.1**	**83.9**	59.5	90.4
YOLOv5-x	88.1	**98.1**	79.6	97.7	76.6	**85.6**	86.0
Ours	**90.2**	94.5	**81.9**	97.1	79.5	71.2	**92.7**

Method	Total	w57	w59	wo
CAB	78.0	79.4	67.5	46.8
MSA_YOLOv3	86.3	85.0	73.0	53.0
YOLOv4	88.4	**96.4**	89.7	**69.1**
YOLOv5-x	88.1	89.0	71.9	55.7
Ours	**90.2**	93.7	**91.0**	68.6

Bold values indicate the best performance in the current category

recognition and intend to collect night-only traffic sign datasets for some attempts.

Funding This work was partially supported by National Key R&D Program of China (No. 2020YFB1600400), Guangzhou Science and Technology Plan Project (No. 202007050004), Shenzhen Fundamental Research Program (No. JCYJ20200109142217397), National Natural Science Foundation of China (Grant Nos. 52172350, U1811463).

Data Availability Statement The public dataset is available from http://cg.cs.tsinghua.edu.cn/traffic-sign/. The source code is also available, by contacting this email: zhangrh25@mail.sysu.edu.cn.

Declarations

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this article.

References

1. Zhang R, He Z, Wang H, You F, Li K (2017) Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system. IEEE Access 5:6649–6660
2. Intelligent recognition system, https://www.qzshangwu.com/t/1/Wap/NewsView1.aspx?id=19583
3. Dinh VQ, Lee Y, Choi H, Jeon M (2018) Real-time traffic sign recognition. pp 206–212
4. Liang M, Yuan M, Hu X, Li J, Liu H (2013) Traffic sign detection by ROI extraction and histogram features-based recognition. pp 1–8
5. Greenhalgh J, Mirmehdi M (2012) Real-time detection and recognition of road traffic signs. IEEE Trans Intell Transp Syst 13(4):1498–1506
6. Yang Y, Luo H, Xu H, Wu F (2015) Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst 17(7):2022–2031
7. Chen X, Li Z, Yang Y, Qi L, Ke R (2020) High-resolution vehicle trajectory extraction and denoising from aerial videos. IEEE Trans Intell Transp Syst 22(5):3190–3202
8. Chen X, Lu J, Zhao J, Qu Z, Yang Y, Xian J (2020) Traffic flow prediction at varied time scales via ensemble empirical mode decomposition and artificial neural network. Sustainability 12(9):3678
9. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection, pp 779–788
10. Ren S, He K, Girshick R, Sun J (2016) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
11. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
12. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. European conference on computer vision. Springer, Cham, pp 740–755
13. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw 32:323–332
14. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C (2013) Detection of traffic signs in real-world images: the German traffic sign detection benchmark. pp 1–8
15. Møgelmose A, Liu D, Trivedi MM (2015) Detection of US traffic signs. IEEE Trans Intell Transp Syst 16(6):3116–3125
16. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. pp 2110–2118
17. Cai Z, Fan Q, Feris RS, Vasconcelos N (2016) A unified multi-scale deep convolutional neural network for fast object detection. European conference on compute vision. Springer, Cham, pp 354–370
18. Yang F, Choi W, Lin Y (2016) Exploit all the layers: fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. pp 2129–2137
19. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, FuC Y, Berg AC (2016) Ssd: Single shot multibox detector. European conference on compute vision. Springer, Cham, pp 21–37
20. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125
21. Fu C.-Y, Liu W, Ranga A, Tyagi A, Berg A. C (2017) Dssd: deconvolutional single shot detector, arXiv preprint arXiv:1701.06659
22. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969
23. Chen S, Wang B, Tan X, Hu X (2018) Embedding attention and residual network for accurate salient object detection. IEEE Trans Cybern 50(5):2050–2062
24. Cui L, Ma R, Lv P, Jiang X, Gao Z, Zhou B, Xu M (2018) Mdssd: multi-scale deconvolutional single shot detector for small objects, arXiv preprint arXiv:1805.07009
25. Bochkovskiy A, Wang C.-Y, Liao H.-Y. M (2020) Yolov4: optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934
26. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: a survey, arXiv preprint arXiv:1905.05055
27. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation, pp 580–587
28. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
29. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE International conference on computer vision, pp. 1440–1448
30. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. Eur Conf Comput Vis. Springer, Cham, pp 21–37
31. Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. pp 2980–2988
32. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE Computer society conference on computer vision and pattern recognition (CVPR'05), pp 886–893. IEEE
33. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
34. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263–7271
35. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement, arXiv preprint arXiv:1804.02767
36. Liu C, Li S, Chang F, Wang Y (2019) Machine vision based traffic sign detection methods: review, analyses and perspectives. IEEE Access 7:86578–86596

37. Gómez-Moreno H, Maldonado-Bascón S, Gil-Jiménez P, Lafuente-Arroyo S (2010) Goal evaluation of segmentation algorithms for traffic sign recognition. IEEE Trans Intell Transp Syst 11(4):917–930
38. Zhang K, Sheng Y, Li J (2012) Automatic detection of road traffic signs from natural scene images based on pixel vector and central projected shape feature. IET Intell Transp Syst 6(3):282–291
39. Salti S, Petrelli A, Tombari F, Fioraio N, Stefano LD (2015) Traffic sign detection via interest region extraction. Pattern Recogn 48(4):1039–1049
40. Fang CY, Chen SW, Fuh CS (2003) Road-sign detection and tracking. IEEE Trans Veh Technol 52(5):1329–1341
41. Barnes N, Zelinsky A, Fletcher LS (2008) Real-time speed sign detection using the radial symmetry detector. IEEE Trans Intell Transp Syst 9(2):322–332
42. Chen T, Lu S (2016) Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression. IEEE Trans Veh Technol 65(6):4006–4015
43. Mogelmose A, Liu D, Trivedi MM (2015) Detection of US traffic signs. IEEE Trans Intell Transp Syst 16(6):3116–3125
44. Park JG, Kim KJ (2013) Design of a visual perception model with edge-adaptive Gabor filter and support vector machine for traffic sign detection. Expert Syst Appl 40(9):3679–3687
45. Berkaya SK, Gunduz H, Ozsen O, Akinlar C, Gunal S (2016) On circular traffic sign detection and recognition. Expert Syst Appl 48:67–75
46. Chen G,Wang H, Chen K, Li Z, Song Z, Liu Y, Chen W, Knoll A (2020) A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal, IEEE Transactions on systems, man, and cybernetics: systems
47. Liu Z, Du J, Tian F, Wen J (2019) MR-CNN: a multi-scale region-based convolutional neural network for small traffic sign recognition. IEEE Access 7:57120–57128
48. Liu Z, Li D, Ge SS, Tian F (2020) Small traffic sign detection from large image. Appl Intell 50(1):1–13
49. Cui L, Lv P, Jiang X, Gao Z, Zhou B, Zhang L, Shao L, Xu M (2020) Context-aware block net for small object detection. In: IEEE Transactions on cybernetics
50. Liu S, Huang D (2018) Receptive field block net for accurate and fast object detection
51. Li J, Liang X, Wei Y, Xu T, Feng J, Yan S (2017) Perceptual generative adversarial networks for small object detection. pp 1222–1230
52. Bai Y, Zhang Y, Ding M, Ghanem B (2018) Finding tiny faces in the wild with generative adversarial network, pp 21–30
53. Fang L, Zhao X, Zhang S (2019) Small-objectness sensitive detection based on shifted single shot detector. Multimedia Tools Appl 78(10):13227–13245
54. Eggert C, Zecha D, Brehm S, Lienhart R (2017) Improving small object proposals for company logo detection, pp 167–174
55. Yu F, Koltun V (2016) Multi-scale context aggregation by dilated convolutions. In: ICLR
56. Yu F, Koltun V, Funkhouser T (2017) Dilated residual networks
57. Jocher G, Stoken A, Borovec J, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, yxNONG, A. Hogan, lorenzomammana, AlexWang1900, A. Chaurasia, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Durgesh, F. Ingham, Frederik, Guilhen, A. Colmagro, H. Ye, Jacobsolawetz, J. Poznanski, J. Fang, J. Kim, K. Doan, L. Yu, ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration, https://doi.org/10.5281/zenodo.4418161 (Jan. 2021)
58. Wang C-Y, Mark Liao H-Y, Wu Y-H, Chen P-Y, Hsieh J-W, Yeh I-H (2020) CSPNeT: a new backbone that can enhance learning capability of CNN. pp 390–391
59. Misra D (2019) Mish: a self regularized non-monotonic neural activation function, arXiv preprint arXiv:1908.08681
60. Liu S, Qi L, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proceedings of the IEEE Conference on computer vision and pattern recognition, pp. 8759–8768
61. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
62. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on computer vision and pattern recognition (CVPR)
63. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
64. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2020) Random erasing data augmentation. AAAI 34(7):13001–13008
65. Zhang H, Qin L, Li J, Guo Y, Xu Z (2020) Real-time detection method for small traffic signs based on Yolov3. IEEE Access 8:64145–64156