Multimedia Tools and Applications (2022) 81:17779–17791
https://doi.org/10.1007/s11042-022-12163-0

Traffic sign recognition based on deep learning

Yanzhao Zhu[1] · Wei Qi Yan[1]

Received: 28 July 2021 / Revised: 15 November 2021 / Accepted: 3 January 2022 /
Published online: 7 March 2022

Abstract
Intelligent Transportation System (ITS), including unmanned vehicles, has been gradually matured despite on road. How to eliminate the interference due to various environmental factors, carry out accurate and efficient traffic sign detection and recognition, is a key technical problem. However, traditional visual object recognition mainly relies on visual feature extraction, e.g., color and edge, which has limitations. Convolutional neural network (CNN) was designed for visual object recognition based on deep learning, which has successfully overcome the shortcomings of conventional object recognition. In this paper, we implement an experiment to evaluate the performance of the latest version of YOLOv5 based on our dataset for Traffic Sign Recognition (TSR), which unfolds how the model for visual object recognition in deep learning is suitable for TSR through a comprehensive comparison with SSD (i.e., single shot multibox detector) as the objective of this paper. The experiments in this project utilize our own dataset. Pertaining to the experimental results, YOLOv5 achieves 97.70% in terms of mAP@0.5 for all classes, SSD obtains 90.14% mAP in the same term. Meanwhile, regarding recognition speed, YOLOv5 also outperforms SSD.

Keywords Deep learning · Traffic sign recognition · CNN · YOLOv5 · SSD

1 Introduction

In recent years, with the outbreak of Artificial Intelligence (AI), the vehicle-aided driving system has updated previous driving mode. By acquiring real-time road condition information, the system promptly reminds drivers to make accurate operations, thereby prevent car accidents due to driver fatigue. In addition to the auxiliary driving systems, development of autonomous vehicles also requires rapid and accurate detection of traffic signs from digital images.

✉ Wei Qi Yan
dcsyanwq@gmail.com

[1] Auckland University of Technology, CBD, Auckland, New Zealand

Traffic Sign Recognition (TSR) is to detect the location of traffic signs from digital images or video frames, given a specific classification [25]. The TSR methods basically make use of visual information such as shape and color of traffic signs. However, the conventional TSR algorithms are facing drawbacks in real-time tests, such as being easily restricted by driving conditions, including lighting, camera angle, obstruction, driving speed, and so on. It's also very difficult to achieve multitarget detection, easy to miss visual objects because of slow recognition [6].

With continuous improvement of computer hardware, the limitation of artificial neural networks has been well alleviated, which has brought machine learning into a golden time of development. Deep learning is a type of machine learning methods [7]. A deep neural network model simulates the neural structure of our human brain while processing information. Using this neural network model to extract the effective features from the road image is much better than the conventional TSR algorithms, which has the potential to improve the robustness and generalization of the algorithms [22].

The research outcomes in TSR not only avoid traffic accidents and protect drivers, but also help inspect traffic signs on roads efficiently and accurately, which reduce unnecessary manpower and resources. In addition, it also provides technical support for unmanned and auxiliary driving. Therefore, the research work based on deep learning has tremendous significance and is invaluable to our daily life.

In this paper, we mainly investigate how to achieve an accurate and real-time TSR model based on deep learning. Our contributions lie in three aspects. Firstly, we collect and augment sample images to form a new dataset for our traffic signs, which contains 2,182 images with eight classes. Secondly, regarding the latest version of YOLOv5, we implement our experiments and evaluate TSR performance based on our dataset. The key metrics and parameters provide a few essential references for further explorations and exploitations. Finally, we conduct a detailed comparison of TSR performance between YOLOv5 and SSD. We also analyze and justify the advantages and disadvantages of these two deep learning models.

We review literature in Section 2, our methods are depicted in Section 3. Our results are showcased in Section 4. Our conclusion and future work will be presented in Section 5.

2 Literature review

TSR has always been a hot research topic in recent years. For this purpose, TSR is investigated to detect traffic sign region and non-traffic sign area in complex scene of images, TSR is to extract the specific features represented through traffic sign patterns [20]. The existing TSR methods are basically grouped into two categories: One is based on traditional methods, the other is related to deep learning methods.

The main steps of TSR methods based on color and shape of a given image are to extract the visual information contained in the candidate area, capture and segment the traffic signs in the image, and correctly label the signs through patter classification [21]. Although TSR requires color and shape information which is employed to improve the recognition accuracy. The problems of illumination changes or color fading of traffic signs, as well as the deformation and the occlusion of traffic signs, are still unresolved problem [14]. Conventional machine learning methods usually selected specified visual features and take use of the features to classify the classes of traffic signs. The specific features include Haar-like features, HOG features, SIFT features, and so on [3].

Conventional TSR methods are based on template matching, which needs to extract and utilize the invariant and similar visual features of traffic signs, the matching algorithms are run for pattern classification. The feature representation of these methods needs to be specified well, which is a tough problem to describe the visual features precisely, because of the variations of traffic signs [17, 24].

The neural networks, Bayesian classifier, random forest, and Support Vector Machine (SVM) are employed as classifiers. However, the performance of conventional machine learning methods depends on the specified features, they are prone to missing the key features. Furthermore, for different classifiers, corresponding feature description information is required. Hence, traditional machine learning methods have limitations, their real-time performance is not comparative relatively.

Deep learning utilizes a multilayer neural network to automatically extract and learn the features of visual objects, which has merits for image processing [29]. CNN models are one of the most popular deep learning approaches for TSR. TSR algorithms are based on region proposals, also known as two-stage detection algorithm, the core idea is selective search [10], its advantages are the great performance of detection and positioning, but the cost is a large amount of computations and high-performance hardware for computing.

The CNN models encapsulate R-CNN, Fast R-CNN, and Faster R-CNN. Faster R-CNN combines the regression of bounding boxes and object classification, takes use of end-to-end methods to detect visual objects, which not only improve the accuracy of object detection, but also uplift the speed of object recognition. The road signs usually were detected from the driver's point of view, in this paper, we view the signs from the viewpoint of satellite images. In [24], guided image filtering was employed for the input image to remove image artefacts such as foggy and haze. The processed image is imported into the proposed networks for model training.

Meanwhile, TSR algorithms based on regression, also known as single-stage detection algorithm [1]. This kind of TSR algorithms eliminate the idea of Region Proposal Network (RPN), and directly perform regression and classification in a network. You Only Look Once (YOLO) and Single Shot MultiBox Detector (SSD) belong to the single-stage category.

Visual object detection consists of two tasks, which are classification and positioning. Before the emerging of YOLOs, these two tasks are different in visual object detection. In the YOLO models, the object detection is simply converted into a regression problem. Furthermore, YOLOs follow an end-to-end structure of neural networks for visual object detection that obtains the coordinates of the predicted bounding boxes, the confidence of the target, and the probability of the class that the target belongs to simultaneously through one image input [18].

In 2020, three YOLO versions had been released, i.e., YOLOv4, YOLOv5, and PP-YOLO [17, 24]. When the YOLOv4 was released, it was considered as the faster and more accurate real-time object detection model, which inherits the Darknet and has obtained a distinct average precision (AP) based on Microsoft COCO dataset while achieved a fast detection speed based on Tesla V100. Compared with YOLOv3, the AP and FPS (i.e., frames per second or video frame rate) have been effectively improved.

YOLOv5 was published in 2020. There is little research outcome on the performance of YOLOv5 for TSR. Nevertheless, an experiment of detecting apples was conducted by using YOLOv5 to compare with the performance of YOLOv3 [11]. The experimental results indicate that YOLOv5 outperformed the previous model. YOLOv5 obtained 4.30% increment of the detection accuracy. Moreover, a similar experiment was conducted for the apple

picking-up [26]. The comparable outcomes with an improved YOLOv5s model, which were 14.95% and 4.74%, are satisfactory by comparing YOLOv3 and YOLOv4, respectively.

SSD is well known since it has been proposed [16]. Meanwhile, the SSD model is already being improved and employed to detect visual object in various fields. Recently, the experiments are implemented based on CTSD dataset with the improved SSD model, the results reach 94.40% for the precision and 92.60% of the recall [9]. Besides, a comparison of traffic sign recognition between SSD and YOLOv2 was carried out [4]. The GTSRB dataset was taken into consideration. In general, SSD was 21.00% less than YOLOv2 in accuracy, the latter was 16.00% faster than the SSD model.

3 Methodology

3.1 YOLOv5

The series of YOLO models have been updated to YOLOv5. The accuracy of visual object detection continues being updated; the regression is always adopted as its core idea. In this experiment, we take the latest version of YOLOv5 as one of the NZ-TSR models. The structure of YOLOv5 algorithm is very similar to that of YOLOv4. The entire network model is divided into four parts: Input, backbone, neck, and the prediction layer. In Fig. 1, the network structure of YOLOv5 is shown in detail.

In the input part, YOLOv5 and YOLOv4 both utilize mosaic method to enhance the input data. The algorithm needs to normalize the input image to a fixed size, the standard size of the image is 608 × 608 × 3. In addition, the network training is based on initial anchor box to obtain prediction box through comparing it with the actual annotated box and updating the network model parameters iteratively [23].

The backbone part contains focus module and CSP module [19]. The key step of the focus model is to compress height and width of the input image through slicing operation. The images are spliced to carry out the integration of image dimensional information (i.e., width and height) into the channel information to increase input channels. On the aspect of CSP module, two branches of CSP module are designed in YOLOv5, which are CSP1_X and

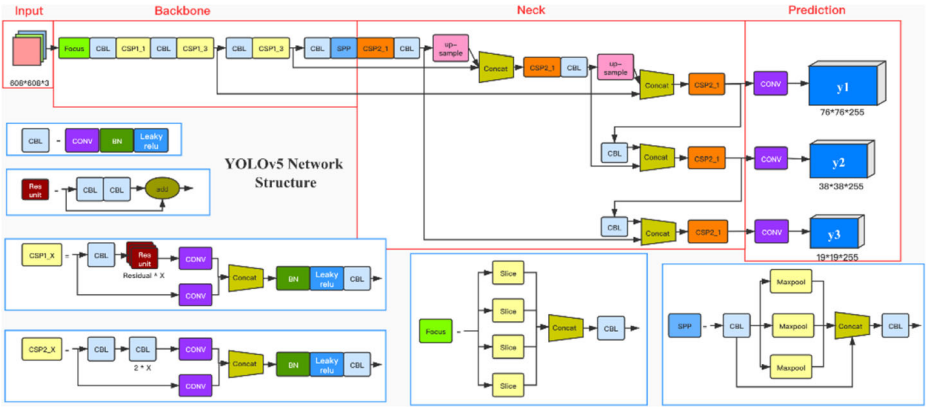

Fig. 1 The diagram of YOLOv5 structure

CSP2_X [13]. Amongst them, CSP1_X module is mainly employed for the backbone network, CSP2_X is mainly taken into use in the neck network.

The neck part in YOLOv5 mimics to YOLOv4, which adopts FPN+PAN structure. Feature Pyramid Network (FPN) is working from top to bottom and utilizes upsampling operation to transfer and fuse information to obtain predicted feature maps [8]. In contrast, PAN (Path Aggregation Network) is a feature pyramid from the bottom to top.

In the prediction part, different from YOLOv4, YOLOv5 makes use of GIoU_Loss as the loss function, which effectively solves the problem if the bounding boxes do not coincide [12]. GIoU is calculated as

$$GIoU = IoU - \frac{|C - (A \cup B)|}{|C|}, \tag{1}$$

where C expresses the smallest box for arbitrary bounding boxes A and B, enclosing A and B. After that, the ratio of the area C is calculated and subtracted from the IoU of A and B. GIoU is treated as a distance. So GIoU loss is derived as

$$GIoU\_Loss = 1 - GIoU = 1 - \left(IoU - \frac{|C - (A \cup B)|}{|C|} \right). \tag{2}$$

3.2 SSD

The SSD model utilizes multiple size detection boxes while extracting object features and generating various feature maps that strengthen the ability of network feature extraction. The SSD model mainly contains two parts as shown in Fig. 2.

The first part is basic feature extraction network, which adopts VGG-16 network without dropout layer, FC8 and softmax classification layers. It replaces the fully connected layers FC6 and FC7 in the ordinary VGG network with convolutional layers Conv6 and Conv7 [16]. In the second part, four convolutional layers of Conv8, Conv9, Conv10, and Conv11 have been newly added. Each convolutional layer utilizes a 1 × 1 convolution kernel for dimensionality reduction and then makes use of a 3 × 3 convolution kernel for feature extraction [27].

The loss function of the SSD model consists of two parts: The localization loss (L_{loc}) and the confidence loss (L_{conf}) [5]. The entire loss function is weighted sum of localization loss and the confidence loss, as shown in Eq. (3).

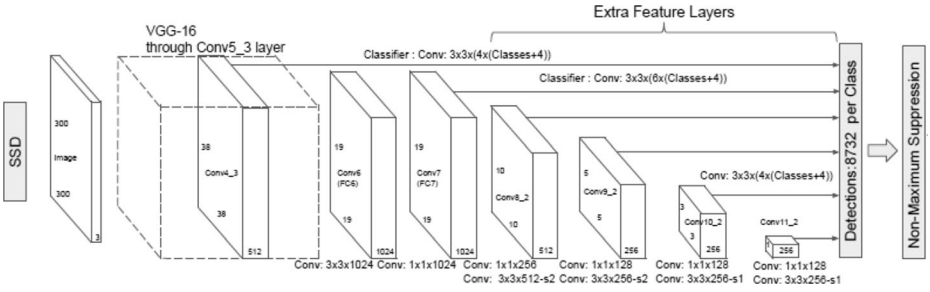

Fig. 2 The diagram of SSD network structure

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right), \tag{3}$$

where N represents the number of positive instances in the prediction box, c is the confidence of the predicted classification, l is the prediction box by using the proposed model, g is labelled box for the ground truth, α is weight coefficient of the localization loss and the confidence loss [28].

The confidence loss function (L_{conf}) adopts softmax loss [2], the input is confidence of each classification c, L_{conf} is presented in Eq. (4).

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log\left(\widehat{c}_i^{p}\right) - \sum_{i\in Neg} log\left(\widehat{c}_i^{0}\right), \tag{4}$$

$$\widehat{c}_i^{p} = \frac{exp(c_i^{p})}{\sum_p exp(c_i^{p})}. \tag{5}$$

The localization loss function (L_{loc}) adopts smooth L_1 loss [28] as the parameters of the prediction box (l) and the labelled box (g) for the ground truth. It also includes the center coordinate position (x, y), width w and height h. So L_{loc} calculation is shown in Eq. (6).

$$smooth_{L1}(x) = \left\{ 0.5x^2, \quad |x| \right. \tag{6}$$

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N}\sum_{m\in\{cx,cy,w,h\}} x_{ij}^{k} smooth_{L1}\left(l_i^m - g_i^m\right), \tag{7}$$

where g_i^m is offset of the labelled box related to the default detection box, l_i^m is prediction box output by the model. Therefore, the prediction box output by the SSD model is not the direct coordinates of the prediction box but the offset of the prediction box is related to the detection box.

4 Experiments

4.1 Data collection

In this experiment, we selected eight classes of traffic signs with high awareness and important safety significance. Because of sparsity of traffic signs on road, we collected traffic sign images instead of driving videos. We split them into two groups, both two groups are captured from the streets of our city by using our mobile cameras.

Our dataset is composed of 2,182 traffic sign images which are labelled as “No U-turn” (271 images), “Road bump” (329 images), “Road works” (294 images), “Watch for children

crossing" (176 images), "Crosswalk ahead" (313 images), "Give way" (317 images), "Stop" (286 images) and "No entry" (196 images), which are shown in Table 1.

4.2 Dataset augmentation

For the raw data in our dataset, a few images were captured in landscape view. Firstly, we made use of the software called *JPEG Autorotate* to rotate the images to portrait direction. After that, due to ultrahigh definition images with too long training time, we resized the image whilst keeping the same aspect ratio between width and height. Thus, we normalized all images in our dataset to be 1128 × 2016 and 1536 × 2048.

The image annotation for model training in YOLOv5 requires the label information. In this paper, we utilized a labelling tool, namely *Labellmg*. Especially, we need to convert the format which suits to *YOLO* because the default format was designed for *PascalVOC*. Each label comprises of five parameters: Index of classification, center point coordinates (x, y), width w, $w \geq x \geq 1$, and height h, $h \geq y \geq 1$. Once all labelling work is accomplished, we group all images in our dataset into training test and test dataset with the proportion 8:2. We put images and corresponding annotation files into our folders, respectively.

Table 1 Our dataset summarization

Class	Sample	Num.	Class	Sample	Num.
No U-turn		271	Road bump		329
Road works		294	Watch for children crossing		176
Crosswalk ahead		313	Give way		317
Stop		286	No entry		196

Compared to YOLOv5, the dataset for training in the SSD model requires the VOC2007 format. Therefore, in *Labellmg*, we adopted the default format. The image labelling information is stored as a *.xml* file in the specified folder. The*.xml* file contains the label classes, coordinates, width, and height. The formal VOC2007 dataset encompasses *Annotations* folder, *ImageSets* folder, *JPEGImages* folder, *SegmentationClass* folder, and *SegmentationObject* folder.

For the SSD dataset in our experiment, we define the sample number in training-validation dataset as 80.00% of the total, the number of test dataset is 20.00%, and the numbers of training and validation datasets are 64.00% and 16.00%, respectively.

4.3 Implementations

In order to implement the TSR experiment with YOLOv5 and SSD based on our own dataset, we took use of Google Collaboratory (Colab) platform with powerful GPU support. The key configuration of our hardware and software as well as the parameters of our experiments is listed in Table 2.

Once the experimental environment is completely set up, we need to mount our Google Drive to Colab and access the prepared dataset. For the experimental parameters, YOLOv5 is shown on the left side in Fig. 3(a), SSD is presented on the right side in Fig. 3(b).

4.4 Experimental results

YOLOv5 model has an excellent visualization function in the result. At first, we visually display its final recognition results for our dataset as shown in Fig. 5. From Fig. 4, we clearly observe TSR in our YOLOv5 experiment is dramatically accurate.

In Table 3, we state the precision of “Road bump”, “Cross walk”, “Give way”, and “No entry”. The lowest precision obtained by “No U-turn” is 0.94. In terms of recall, the values for almost eight classes are all over 90.00%, which indicates the excellent TSR performance of YOLOv5 in our dataset. Therefore, undoubtedly, for the mean average precision, “No U-turn” obtains as high as 99.50%, all other classes are around 97.00%. It demonstrates that the YOLOv5 model is able to achieve the completely accurate prediction of NZ-TSR in our dataset.

In the end, all specific evaluation metrics of YOLOv5 in our dataset are shown in Fig. 5. Especially, the second and third columns are the mean values of the loss functions of the training dataset and the validation dataset for visual object detection and classification, respectively. The smaller the value is, the better the recognition performance of the model will be.

Table 2 The key configuration and environment parameters of our experiments

Operation System	Ubuntu 18.04.5 LTS
GPU	Tesla P100-PCIE-16GB
RAM	26GB
Programming Language	Python 3.7.10
CUDA	Version 11.0.228
PyTorch	Version 1.8.1

Network	YOLOv5x
Image size	640×640
Batch size	16
Epochs	200

(a)

init_epoch	0
freeze_epoch	100
unfreeze_epoch	200
batch_size	16
input_shape	(300, 300, 3)
confidence	0.5
num_iou	0.45
cuda	True

(b)

Fig. 3 YOLOv5 vs. SSD experimental parameters (**a**) YOLOv5 (**b**) SSD

Usually, the most convenient and direct way to evaluate the experiment results is accuracy. In this paper, we make use of PR curves to demonstrate the tradeoff between precision rates and recall rates of the TSR performance of the models.

In Fig. 6, we see mAP@0.5 results for eight classes in the SSD experiment. Overall, the accuracy of TSR in almost all classes reaches nearly 90.00%. In particular, the TSR of "Give way" has the best performance, the final average precision is as high as 97.06%. However, the average precision of "Watch for children crossing" is quite low, only 78.32%. The reason is highly remarked that the number of instances of that specific class is lower than others.

All SSD experimental results are summarized in Table 4. The results illustrate the SSD model has a relatively good outcome apart from "Watching for children crossing". In other words, for the SSD model, the more instances the dataset contains, the more accurate prediction will be.

4.5 Comparisons

After conducting a comprehensive comparison of YOLOv5 and SSD in our dataset, we see intuitively that the mean average precision of all eight classes obtained by YOLOv5 and SSD

Fig. 4 TSR results by using YOLOv5

Table 3 The YOLOv5 experimental results with our dataset

Classes of traffic signs	Precisions	Recalls	mAP@0.5
No U-turn	0.937	1.000	0.995
Road bump	1.000	0.903	0.965
Road works	0.979	0.897	0.928
Watch for children crossing	0.997	1.000	0.995
Crosswalk ahead	1.000	0.979	0.986
Give way	1.000	1.000	0.995
Stop	0.984	0.938	0.975
No entry	1.000	0.938	0.976

is 0.98 and 0.90, respectively. From the perspective of accuracy rate of each class, the experimental results of YOLOv5 are all better than SSD except for "Road works". Furthermore, the performance of YOLOv5 for "No U-turn", "Watch for children crossing" and "Give way" is remarkable, which has reached 0.99, close to 100.00% detection. For the SSD experiment, the highest recognition accuracy is 0.98 obtained in "Road works". However, in "Watch for children crossing" with a small number of samples, the accuracy is only 0.78. In the end, on the aspect of the TSR accuracy in our dataset, both YOLOv5 and SSD show good capabilities, but YOLOv5 performs a bit better.

For the TSR efficiency, with the same number of images in the test dataset, YOLOv5 spends only 15 s, while SSD needs 129 s. The speed of TSR by using YOLOv5 is 30 FPS, nearly ten times faster than SSD, which is 3.49 *fps*. Therefore, YOLOv5 outperforms SSD as well in terms of TSR efficiency.

5 Conclusion and future work

This project aims to probe the accuracy and speed of TSR based on the dataset of our traffic signs. Hence, in this paper, we selected the latest version of the series of YOLO algorithms, namely YOLOv5, to evaluate its performance. Besides, we also identify which model is much suitable for the TSR between YOLOv5 and SSD. In this experiment, we adopt a customized

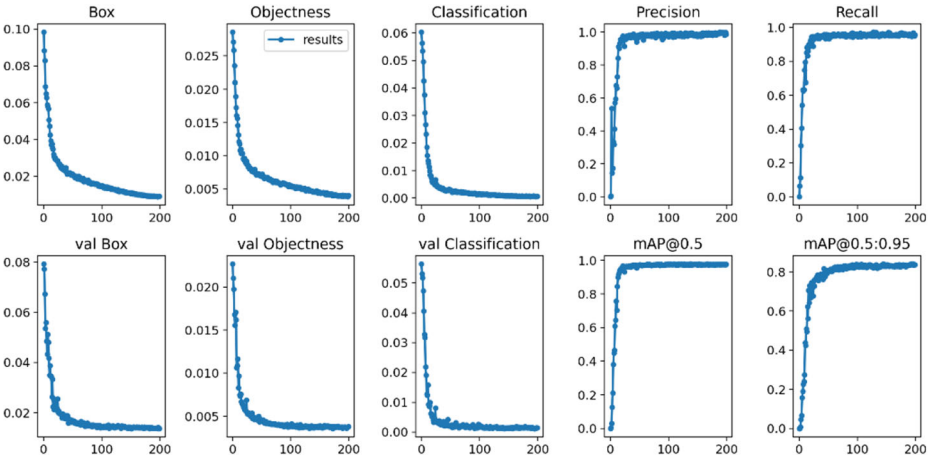

Fig. 5 All specific evaluation metrics of YOLOv5 in our dataset

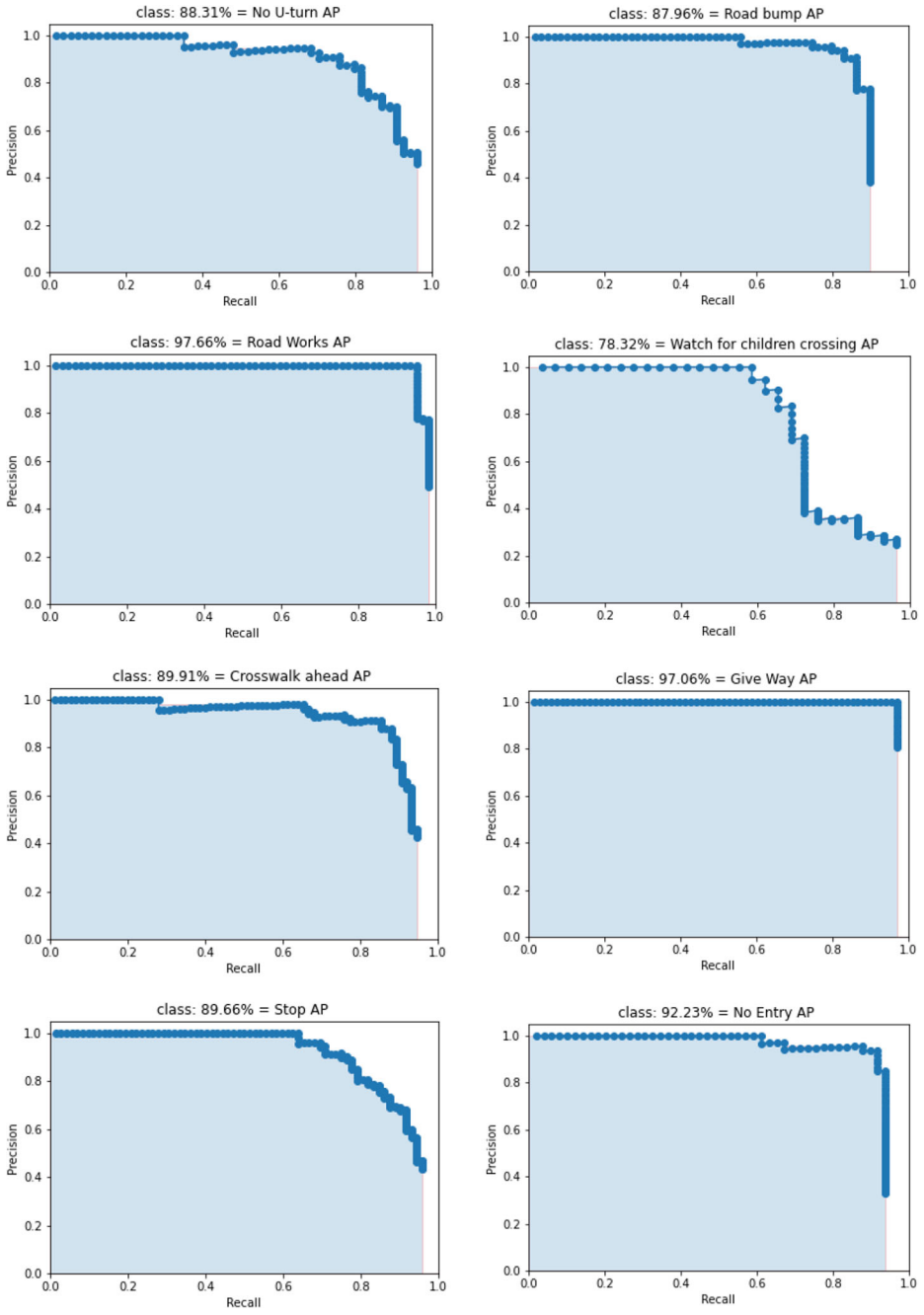

Fig. 6 Precision-recall curves for eight classes in the SSD experiments

dataset of our traffic signs, which contains 2,182 traffic sign images including eight classes. Then, we implement a well-designed experiment based on the Google Colab platform having a

Table 4 The SSD experimental results with our dataset

Classes	Precisions	Recalls	mAP@0.5
No U-turn	0.875	0.778	0.881
Road bump	0.895	0.864	0.880
Road works	0.967	0.952	0.977
Watch for children crossing	1.000	0.414	0.783
Crosswalk ahead	0.880	0.880	0.899
Give way	1.000	0.956	0.971
Stop	0.912	0.722	0.897
No entry	0.935	0.878	0.922

very strong computational capability. In addition, we also analyze and compare the performance of the two models by using our evaluation metrics.

From the experimental results, the accuracy of YOLOv5 is up to 97.70% for all classes, the mean average precision in each class is over 90.00%. Hence, SSD obtains 90.14% on the accuracy in general. But for the class with fewer samples, it only has 78.32% recognition rate. Therefore, YOLOv5 performs better than SSD in terms of recognition accuracy. Furthermore, from the perspective of recognition speed, YOLOv5 is faster than SSD with 30 *fps* (frames per second), SSD only has 3.49 *fps*. We believe that YOLOv5 is more suitable for TSR in real-time traffic environment.

In future, we will keep extending our datasets to cover all classes of our traffic signs. Meanwhile, more newly developed models for visual object recognition, such as Mask R-CNN, CapsNet, and Siamese neural network would be included. Capsule neural network (CapsNet) has been employed for effectively identifying a class of traffic signs which have spatial relationships. Compared with the well-known deep neural networks, capsule networks tackle the topological relationship between visual objects. In addition, we will adopt professional evaluation metrics to assess the performance of our models from multiple aspects in future [15].

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations This work has not any funding support, it has not any conflicts of interests or competing interests.

References

1. Bangquan X, Xiong WX (2019)Real-time embedded traffic sign recognition using efficient convolutional neural network. IEEE Access 7:53330–53346

2. Chen Q, Huang N, Zhou J, Tan Z (2018) An SSD algorithm based on vehicle counting method. Chinese Control Conference
3. Ellahyani A, Ansari M, Lahmyed R, Trémeau A (2018) Traffic sign recognition method for intelligent vehicles. J Opt Soc Am 35(11):1907–1914. https://doi.org/10.1364/JOSAA.35.001907
4. Garg P, Chowdhury DR, More VN (2019) Traffic sign recognition and classification using YOLOv2, Faster R-CNN and SSD. International Conference on Computing, Communication and Networking Technologies
5. Hao G, Yingkun Y, Yi Q (2019) General target detection method based on improved SSD. IEEE Joint International Information Technology and Artificial Intelligence Conference
6. He Z, Nan F, Li X, Lee SJ, Yang Y (2020) Traffic sign recognition by combining global and local features based on semi-supervised classification. IET Intel Transport Syst 14(5):323–330
7. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
8. Hu GX, Hu BL, Yang Z, Huang L, Li P (2021) Pavement crack detection method based on deep learning models. Wirel Commun Mob Comput 2021. https://doi.org/10.1155/2021/5573590
9. Huo A, Zhang W, Li Y (2020) Traffic sign recognition based on improved SSD model. International Conference on Computer Network, Electronic and Automation
10. Jin Y, Fu Y, Wang W, Guo J, Ren C, Xiang X (2020)Multi-feature fusion and enhancement single shot detector for traffic sign recognition. IEEE Access 8:38931–38940
11. Kuznetsova A, Maleva T, Soloviev V (2020) Detecting apples in orchards using YOLOv3 and YOLOv5 in general and close-up images. International Symposium on Neural Networks
12. Li S, Gu X, Xu X, Xu D, Zhang T, Liu Z, Dong Q (2021) Detection of concealed cracks from ground penetrating radar images based on deep learning algorithm. Constr Build Mater 273:121949
13. Lian J, Yin Y, Li L, Wang Z, Zhou Y (2021) Small object detection in traffic scenes based on attention feature fusion. Sensors 21(9):3031
14. Lim K, Hong Y, Choi Y, Byun H (2017)Real-time traffic sign recognition based on a general purpose GPU and deep-learning. PLoS One 12(3):e0173317
15. Liu X, Yan W (2021)Traffic-light sign recognition using capsule network. Multimed Tools Appl 80:15161–15171
16. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) SSD: Single shot multibox detector. European Conference on Computer Vision
17. Qin Z, Yan W (2021)Traffic-sign recognition using deep learning. International Symposium on Geometry and Vision (ISGV). Springer, Berlin, pp 13-25
18. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You Only Look Once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition
19. Shi X, Hu J, Lei X, Xu S (2021) Detection of flying birds in airport monitoring based on improved YOLOv5. International Conference on Intelligent Computing and Signal Processing
20. Sun W, Hongji D, Nie S, He X (2019) Traffic sign recognition method integrating multilayer features and kernel extreme learning machine classifier. Comput Mater Continua 60(1):147–161
21. Wang C (2018) : Research and application of traffic sign detection and recognition based on deep learning. International Conference on Robots &; Intelligent System
22. Wu Y, Qin X, Pan Y, Yuan C (2018) Convolution neural network based transfer learning for classification of flowers. IEEE International Conference on Signal and Image Processing
23. Xiaoping Z, Jiahui J, Li W, Zhonghe H, Shida L (2021) People's fast moving detection method in buses based on YOLOv5. Int J Sens Sensor Netw 9(1):30
24. Xing J, Yan W (2021) Traffic sign recognition using guided image filtering. International Symposium on Geometry and Vision (ISGV), Springer, Berlin, pp 85-99
25. Xu S, Niu D, Tao B, Li G (2018) Convolutional neural network based traffic sign recognition system. In International Conference on Systems and Informatics (ICSAI), pp 957-961
26. Yan B, Fan P, Lei X, Liu Z, Yang F (2021) A real-time apple targets detection method for picking robot based on improved YOLOv5. Remote Sensing 13(9):1619
27. Yao Y, Yang Y, Su X, Zhao Y, Feng A, Huang Y, Pu H (2019) Optimization of the bounding box regression process of SSD model. International Conference on Computer Engineering, Information Science & Application Technology
28. Yu G, Fan H, Zhou H, Wu T, Zhu H (2020) Vehicle target detection method based on improved SSD model. J Artif Intell 2(3):125
29. Zhang J, Hui L, Lu J, Zhu Y (2018)Attention-based neural network for traffic sign detection. International Conference on Pattern Recognition