SN Computer Science (2022) 3:461
https://doi.org/10.1007/s42979-022-01355-y

ORIGINAL RESEARCH

The Improved Framework for Traffic Sign Recognition Using Guided Image Filtering

Jiawei Xing[1] · Minh Nguyen[1] · Wei Qi Yan[1]

Received: 2 April 2021 / Accepted: 1 April 2022 / Published online: 30 August 2022

Abstract

In the lighting conditions such as hazing, raining, and weak lighting condition, the accuracy of traffic sign recognition is not very high due to missed detection or incorrect positioning. In this article, we propose a traffic sign recognition (TSR) algorithm based on Faster R-CNN and YOLOv5. The road signs were detected from the driver's point of view and the view was assisted by satellite images. First, we conduct image preprocessing by using guided image filtering for the input image to remove noises. Second, the processed image is input into the proposed networks for model training and testing. Three datasets are employed to verify the effectiveness of the proposed method finally. The outcomes of the traffic sign recognition are promising.

Keywords Traffic sign recognition · Faster R-CNN · GTSDB dataset · FRIDA database

Introduction

Various types of traffic signs have been applied to assist road users. Figure 1 shows a rich assortment of traffic signs which have been set on the roadside. However, only using our human visual systems is tough to eye these signs due to fast moving or weather conditions. Therefore, advanced driver assistance systems have become the focus of our attention [1–3].

At present, traffic sign recognition algorithms have achieved satisfactory results [4, 5], however these algorithms mainly aim at digital images of traffic signs acquired under ideal weather conditions. Due to environmental changes in recent years, haze weather has increased very often that leads to image blur, which in turn slash the recognition accuracy of these algorithms. In response to this issue, an accurate locating and recognition algorithm for traffic signs in haze weather is proposed in this article.

Traffic sign recognition (TSR) was developed in the early 1980s and has been taken a great step in the field of autonomous vehicles in 1987 [6]. It mainly targets speed limit signs and takes use of classic algorithms based on image segmentation as well as template matching. The recognition process takes around 0.50 s on average. Due to the hardware being developed at that time, the systems were not working in real time, the images were relatively small and cannot be integrated into real applications.

Since the 1990s, with the continuous improvement of the hardware and its computing capability, advanced technology in the world has emerged to take effects on discovering the principle of TSR. A variety of solutions have been proposed, such as edge extraction, color-based segmentation, feature vector extraction, artificial neural network, etc. In recent years, with the successful applications of deep learning [7, 8], such as speech recognition, semantic segmentation, etc., deep learning methods have been gradually brought into TSR.

The existing algorithms of traffic sign recognition generally have two key steps: Traffic sign positioning and recognition. Because of the swift development of deep learning, in this paper, our objective is to identify traffic signs from wild weather, thus we propose a deep learning method for TSR based on the Faster R-CNN model.

The rest of the paper is arranged as follows: The existing work is critically reviewed in section "Literature Review".

This article is part of the topical collection "From Geometry to Vision: The Methods for Solving Visual Problems" guest edited by Wei Qi Yan, Harvey Ho, Minh Nguyen and Zhixun Su.

✉ Wei Qi Yan
wyan@aut.ac.nz

[1] Auckland University of Technology, No. 2-14, Wakefield Street, Auckland 1010, New Zealand

Fig. 1 Traffic signs in a foggy weather

The proposed methods of this paper will be detailed in section "Our Methodology". The experimental results will be showcased and analyzed in section "Our Results and Discussion". Our conclusion and our future work will be presented in section "Conclusion".

Literature Review

Traffic sign recognition has become a hot topic in current research. With the progress of hardware, there are various ways to obtain traffic sign images. In terms of image acquisition methods, there are mainly two-fold: One is road condition and traffic information taken by an optical camera on the ground; the other is high-resolution remote sensing image obtained by using satellite transmitting electromagnetic waves to the ground in space, the road signs on the ground are also obtained from these images. Then, deep learning algorithms are proffered to extract visual features from the acquired images to realize road target detection.

A comprehensive scheme [9] was propounded for traffic sign recognition. First, a cascade of trained classifiers was employed to scan the background quickly so as to locate a region of interest (ROI), then Hough transform was applied to shape detection. This method was evaluated based on an image database including 135 traffic signs. The average recognition speed was 25.00 frames per second, the recognition accuracy was 93.00%. Edge detection [10] was accomplished by using a combination of color filtering and closed curves. Through a neural network, the extracted features were applied to classify the targets. The average recognition rate was up to 94.90%. The nearest neighbors were applied to classify and recognize traffic signs from digital images by calculating Euclidean distance between a traffic sign and its standard template, then the image was classified according to the minimum distance.

Girshick et al. [11] proposed a rich feature hierarchical structure for precise target detection and semantic segmentation, Region CNN (R-CNN) uses selective search (SS) [12, 34] instead of traditional. The sliding window method extracted 2,000 target candidate regions on the given image, then took the use of a deep convolutional network to classify the target candidate areas. However, because it performed convolution operations on each candidate area instead of sharing calculations, the detection speed was slow, but with 47.90% segmentation accuracy. He et al. [13] proposed the spatial pyramid pooling network (SPPNets), which improved the speed by sharing convolutional feature maps. Fast R-CNN [14] extracted convolutional feature maps, the training process improved the detection accuracy and speed.

Single shot multibox detector (SSD) [15] was set forth to detect traffic signs using Inceptionv3 network instead of VGG-16 [35]. Pertaining to SSD [37], a random center point with a prior designed strategy was proposed. Douville, et al. [16] firstly normalized the image of traffic signs, then extracted Gabor features, and finally a three-layer perceptron was employed to classify and recognize the traffic sign. A perceptual confrontation network was put forward for highway traffic sign detection [17], which combined Faster R-CNN with a generative confrontation network. The residual network was applied to learn the differences between the feature maps of small visual objects and large target objects so as to uplift the rates of highway traffic sign recognition (HTSR). The detection results have been achieved based on the Tsinghua-Tencent 100 K dataset.

With the development of satellite remote sensing, traffic target detection has been probed based on satellite remote sensing images. In the early stage, a large number of researchers realized target recognition of satellite remote sensing images based on traditional methods. Huang et al. [18] implemented road extraction from remote sensing images according to geometric, radiation and topological features of roads, and classifies them by SVM (i.e., support vector machine) method. The method of the decision tree classifier is related to recursive segmentation of the input image. Its branches represent different segmentation paths and leaves represent the final classification results. Therefore, the whole tree is the process of segmentation.

Eikil and Aurdal [19] proposed vehicle detection based on high-resolution satellite images. Firstly, a rule-based method was employed to segment the image into the normal region and shadow region. Then the targets were classified by using a statistics-based method and the detection results were compared with the results of manual identification. The experimental results show that the image resolution was low and it was difficult to classify objects manually, the detection results of the algorithm are good and close to the results of artificial classification.

Leitloff et.al [20] took use of a Haar-like feature-based AdaBoost algorithm to identify vehicles, combined with a line detection method to find individual vehicles in the fleet. Compared with the method based on statistics alone, the accuracy of this method was improved up to 80.00%. Although the traditional method has achieved good results in target recognition based on satellite remote sensing images, which need to extract features manually, the design process is complicated and lacks good robustness for the diversity of targets.

With the rapid development of deep learning, a breakthrough has been made in the field of pattern recognition. A large number of experts have begun to study the target detection of satellite remote sensing images based on deep learning. Audebert et al. [21] proffered a completely symmetric convolution neural network to obtain more details on shallow layer information, realized the semantic segmentation task of high-resolution remote sensing image. Volpi et al. [22] put forward a multi-path deconvolution method to obtain more low-level details and judge the edge of the object more accurately. As a new field, there are still a slew of problems in using this method for object recognition.

Sherrah et al. [23] utilized general images to pre-train FCN (i.e., fully connected network) and then applied it to remote sensing images, which effectively improved the accuracy of object recognition in remote sensing images. Cheng et al. [24] proposed a multitarget detection framework: Rotation-invariant convolutional neural network (RICNN), which effectively detected a variety of targets in remote sensing images and is a stable and high-performance detection framework. However, the average accuracy of the RICNN method for all objects was only 72.60% on average, and the detection accuracy of different types of objects varied. There are a lot of small-size targets in remote sensing image, which is very difficult to identify, and it is a very challenging part of target detection from remote sensing images. Therefore, deep learning is possible to be applied to the identification of traffic signs from satellite images, and there should be a large room for further development.

Our Methodology

We see the current work has the following defects, ground angle images are influenced by using the environment, the angle, light intensity, the concentration of the mist has an impact on the results, such as the influence on the image is the largest one, we mainly aim at the TSR with haze weather. Our idea for TSR in this paper is depicted in Fig. 2. We first employ digital image processing to cope with foggy images, then input the preprocessed images into a neural network for object detection and classification.

Guided Image Filtering

Image defogging is an important process for haze removal, which enhances visual effects such as edges and contours. There are generally two types of image defogging algorithms, one is histogram equalization, which simply enhances the contrast of the image. The other is an image restoration-based defogging algorithm [25], which takes the use of original images to compare with the foggy images so as to reconstruct the new image. The dehazing result is prominent, but it is difficult to achieve the quality of the original image.

Image filtering is able to resolve the drawbacks of the two dehazing algorithms. The algorithm adopts an image to guide and filter the target image so that the final output image roughly resembles the target image, the texture is akin to the guiding image. The guiding or reference image is either a different one or the same one as the input image itself. If the guiding image is equivalent to the input image, the filtering becomes an edge-preserving operation, which is able to be used for image reconstruction. By using visual features of the guided image filtering, haze image processing for traffic signs achieves the results of image denoising, image smoothing, and fog removal. Therefore, we define the original image as p_i, l_i as the guiding image, and q_i as the output image. The relationship is linear as shown in Eq. (1).

$$q_i = a_k I_k + b_k i \in \omega_k \tag{1}$$

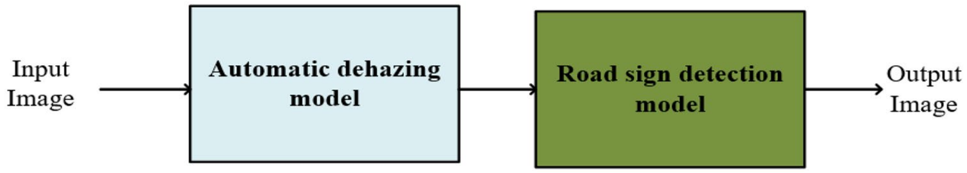


Fig. 2 The pipeline for TSR

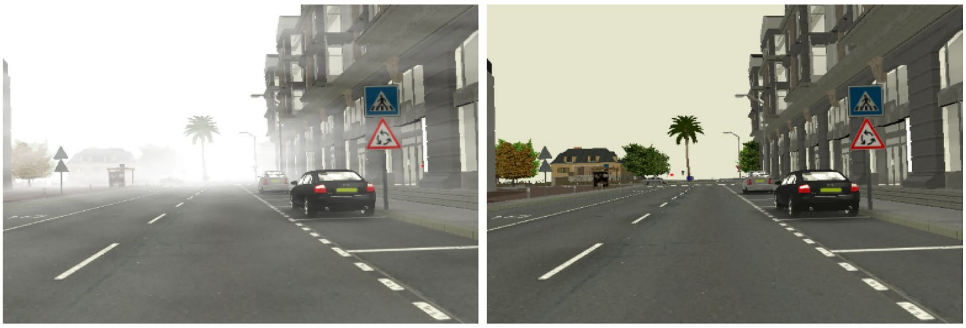

(a) original picture **(b)** after the guided filtering

Fig. 3 **a** The original picture from FROSI databases. **b** The picture obtained by removing the foggy after the guided filtering

where a_k and b_k are specific factors, ω_k is a square window with a centre point k, $i \in \omega_k$ guarantees that a_k is not too big. To ensure the guided image filtering has the best outcome, the difference between the original image and the output image needs to be minimized. Therefore, the cost function $E\ (a_k, b_k)$ is defined as.

$$E\left(a_k, b_k\right) = \sum_{i,k\in\omega_k}\left(\left|\left(q_i - p_i\right)^2 - \varepsilon {a_k}^2\right|\right) \quad (2)$$

The output is the best one if $E(a_k, b_k)$ is the smallest one. We find the least square method by using a_k and b_k,

$$a_k = \frac{\frac{1}{|\omega|}\sum_{i\in\omega_k} I_i q_i - u_k \overline{p_k}}{\sigma_k^2 + \varepsilon}, b_{k=}\overline{p_k} - a_k u_k \quad (3)$$

where u is the mean of I in $\boldsymbol{W}$, σ is the variance of I in W, w is the number of pixels in the window. We input a_k and b_k into Eq. (1) and obtain,

$$q_i = \frac{1}{|\omega|}\sum_{i\in\omega_k}\left(a_k I_k + b_k\right) = \overline{a_i} I_i + \overline{b_i}. \quad (4)$$

Improved Faster R-CNN

Convolutional neural networks (CNNs) usually include a convolutional layer and a pooling layer, where the convolutional layer is normally employed to extract visual features from the target. The feature extraction network in Faster R-CNN is based on a convolutional neural network, which takes use of CNN and rectified linear unit (ReLU) activation function to extract the features from the target image, the extracted features are input into the RPN layer and ROI pooling layer, respectively.

Conventional methods may be the use of sliding windows or selective search to generate detection windows. Faster R-CNN chooses RPN (i.e., region proposal network) to generate the detection window. The network takes advantage of the softmax function to determine the properties of anchor points (foreground or background). Then regression is employed to correct it. Finally, accurate proposals will be obtained.

In Fig. 3, the RPN structure is framed by dotted lines. After 3 × 3 convolution, the feature map flows into two different channels, respectively. The upper one is classified by using the softmax layer to obtain foreground and background. To obtain a relatively accurate proposal, the feature passes through the channel to calculate the offset of the regression. Finally, whilst removing the proposal that exceeds the boundary and the value is too small, the previous information is integrated to obtain a new proposal. With the network structure, the RPN layer basically completes the operation of locating the target.

The input of ROI pooling layer is the proposals of different sizes. However, the input and output sizes of a convolutional neural network after training are fixed, which resizes the proposals to the same.

In Faster R-CNN, we have fine-tuned parameters, set the learning rate to 0.01, the momentum as 0.90, the batch size as 24, and the epoch as 200. The input features contain the proposal of the classification network which is composed of a fully connected layer and softmax activation function so as to attain the predicted probability of each class the traffic sign belongs to. Faster R-CNN is shown in Eq. (5).

$$L\left(\left(f_i\right),\left(l_i\right)\right) = \lambda \frac{1}{N_{\text{reg}}}\sum_{i} f_i^* L_{\text{reg}}\left(l_i, l_i^*\right) + \frac{1}{N}\sum_i L_{cls}\left(f_i, f_i^*\right) \quad (5)$$

where i represents the anchor index, f_i stands for the output probability of the softmax layer of positive samples, f^* means the corresponding prediction probability, l refers to the predicted bounding box, $l*$ denotes the GT (i.e., ground truth) box corresponding to the positive anchor.

Taken into account the advantages of Faster R-CNN, this paper adopts the Faster R-CNN model to detect traffic signs. Faster R-CNN takes advantage of VGG net [26] as the backbone of the net. However, as the basic network improves, in this paper, we take use of GoogLeNet [27] for feature extraction in our experiments. The network parameters are shown in Table 1.

Table 1 The parameters of GoogLeNet

Layers	Types	Sizes	Strides
1	Conv	(7,7)	2
2	Max pooling	(3,3)	2
3	Conv	(3,3)	1
4	Max pooling	(3,3)	2
5	Inception(a)		
6	Inception(b)		
7	Max pooling	(3,3)	2
8	Inception(a)		
9	Inception(b)		
10	Inception(c)		
11	Inception(d)		
12	Inception(e)		
13	Max pooling	(3,3)	2
14	Inception(a)		
15	Inception(b)		

After experimental verification, GoogLeNet has achieved the best results in terms of time-consuming and model performance based on the given dataset. During convolution, the kernels of various sizes were taken for the convolutional operations, the output feature maps are connected together.

Because the traffic sign will show multiple scales in the given image, after feature extraction, the traffic signs with different scales are represented as features. We use cross-layer connections to improve the performance of multiscale target detection.

The detection net that we designed for the cross-layer connection is shown in Fig. 4. As shown in Fig. 4, CNN is accommodated to extract the features of the entire image. The RPN network is applied to extract a series of candidate regions based on the feature map. The change lies in the feature composition of the candidate region. This feature is no longer extracted by using only a single convolution layer but is a fusion of features extracted from multiple convolution layers. The fused features contain not only semantic information but also local information.

In the given dataset, we often find a slew of objects that resemble highway traffic signs. This will generate false detections. To achieve the purpose of reducing false detection, we make use of sample mining [28]. Firstly, the model is used to test the training set. If there are negative samples with a score 0.80 or more in the obtained test results, they will be classified into a new sample class. In this way, the training set contains two classes: Traffic signs and traffic-like objects. The training set is obtained by mining negative samples so as to retrain a new detection model. Traffic signs are classified into classes and added to the training set so that the model has the difference between the two classes during the training time. This resolves the problem that the model cannot classify background objects with minor differences between the positive class and the positive class if the amount of data is insufficient, thereby we obtain a satisfactory outcome (Fig. 5).

Improved YOLOv5

You Only Look Once (YOLO) is a fast and compact open-source object detection model. Compared with other nets, it has stronger performance at the same size and has excellent stability. The YOLO framework treats target detection as a regression problem, it is the first one that the end-to-end net is employed to predict the class and bounding box of

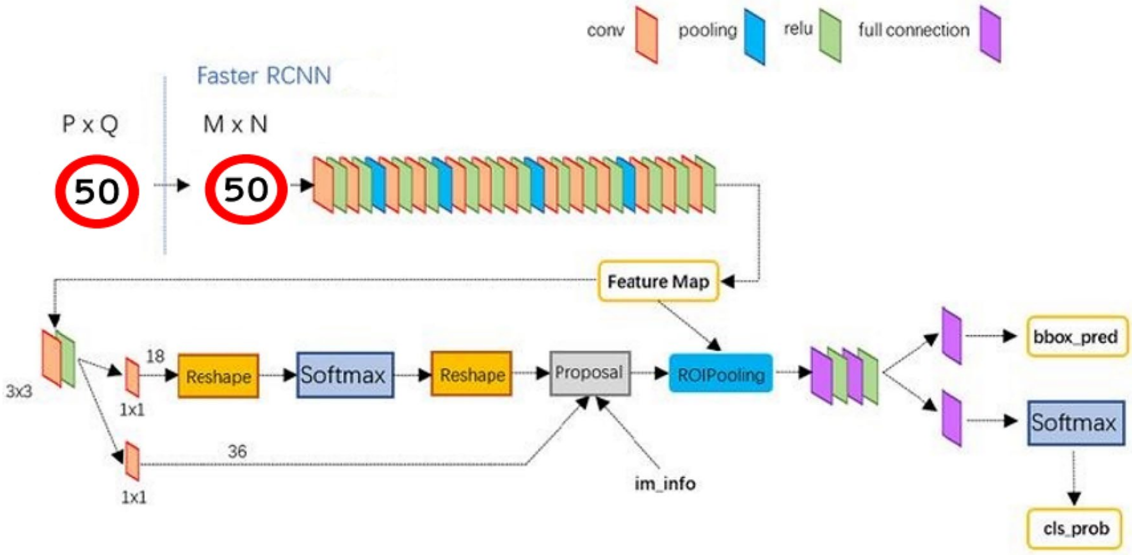


Fig. 4 The structure of Faster R-CNN

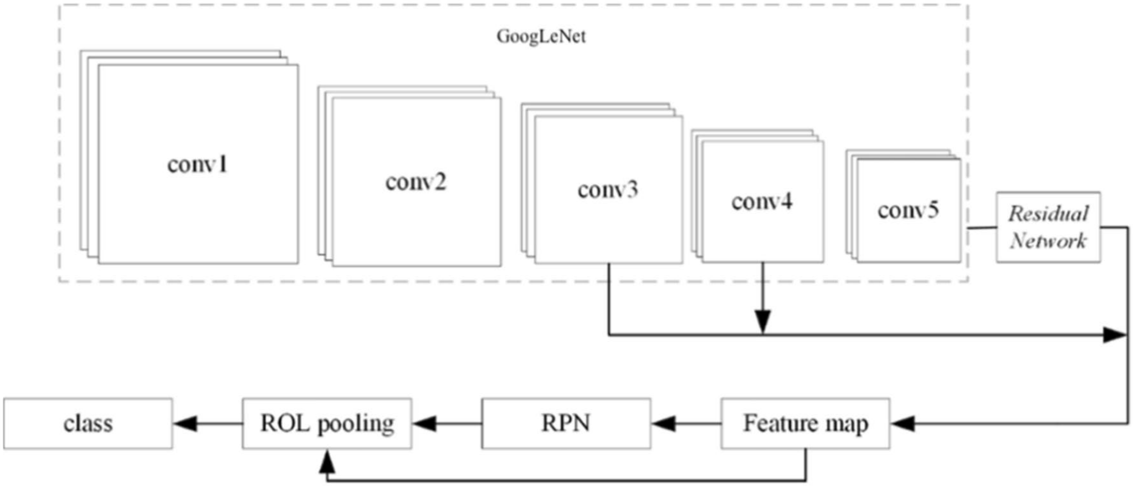


Fig. 5 The improved framework

the visual object. At present, YOLOv5 has a faster recognition speed and smaller network size than YOLOv4. While model training with various datasets, YOLOv3 and YOLOv4 need a separate program to calculate the initial anchor box, YOLOv5 embeds this function into the code to automatically calculate the best anchor box for different datasets. In YOLOv5, we have fine-tuned parameters, set the learning rate as 1.20×10^{-3}, the momentum as 0.95, the batch size as 16, and the epoch as 200 according to the batch size. However, in practice, it was found that the clustering results are deviated from the statistical results of the samples, which affected the performance of subsequent tests. Therefore, in this paper, we optimize the prior anchor box regression algorithm and add the random correction processing of the clustering algorithm.

$$W_b = \mathrm{O}_2^3\left(\mathrm{random}\left[\mathbf{v}_1, \mathbf{v}_2\right] \times w_b\right) \tag{6}$$

where $O_2{}^3$ (●) means that two of every three cluster centres are randomly selected for correction, w_b is the width of prior anchor point before correction, W_b is the width after correction. The numbers reflect the width and the height of the anchor box, respectively.

It is observed that the minimum aspect ratio of the clustering results is 0.53 and the maximum is 0.71. However, for the dataset in this article, the aspect ratio of 70.00% of training samples is between 0.72 and 1.00, 20.00% of samples are between 0.60 and 0.70, 10.00% of samples are between 0.60 and 0.70. From the analysis, we see that there is a deviation between the clustering results and the statistical results.

Compared with pedestrians and vehicles, the physical size of traffic signs is smaller and there are three kinds of traffic signs in most samples. Because the ratio of foreground to background is severely unbalanced, most of the bounding boxes do not contain the target if the one-stage target detector is applied. Because the confidence error of these untargeted bounding boxes is relatively large, the loss of the foreground is submerged in the loss of the background. Therefore, in this paper, we optimize on the basis of the original loss function. The main idea of optimization is to adaptively balance the loss of foreground and background. The loss function encapsulates two parts, namely, regression loss and classification loss.

$$\begin{aligned}\mathrm{loss} = {} & \sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B} E_{ijk}^{obj}\left\{\omega_{\mathrm{coord}}\left[(x_{gt}-x_p)^2+(y_{gt}-y_p)^2+\left(\sqrt{w_{gt}}-\sqrt{w_p}\right)^2+\left(\sqrt{h_{gt}}-\sqrt{h_p}\right)^2\right]\right\}+ \\ & \sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B}\left\{\left[\omega_{obj}E_{ijk}^{obj}(C_{gt}-C_p)^2\right]+\left[\omega_{noobj}E_{ijk}^{noobj}C_p(C_{gt}-C_p)^2\right]\right\}+\sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B}(P_{gt}-P_p)^2\end{aligned} \tag{7}$$

where S is the width and height of the feature map. There are three sizes of the feature map in this article: 52×52, 26×26, 13×13, B is the number of a priori boxes at each anchor point position; E_{ijk}^{obj} represents the anchor point whether the box is responsible for predicting the target, E_{ijk}^{noobj} means not responsible for predicting the target; $x_{gt,}$ $y_{gt,}$ w_{gt}, and h_{gt} are ground truths, x_p, y_p, w_p, and h_p are predicted values, which indicate the coordinates of the object and its width as well as height (in pixels); C_{gt} and C_p represent true value confidence and prediction confidence, respectively; P_{gt} and P_p show classification true value probability and classification prediction probability, respectively; ω is the weight coefficient of each loss part, for weight. The value is set in this paper as $\omega_{\text{coord}} = 5.00$, $\omega_{obj} = 1.00$, $\omega_{noobj} = 0.50$, the purpose of this setting is to reduce the loss of non-target areas and increase the loss of target areas; to further avoid the loss of background values to confidence. In this paper, C_p is employed as a part of the weight to adjust the loss value of the background frame adaptively.

Visual Object Detection from Satellite Images Using YOLOV5

YOLOv5 has high flexibility and productivity owing to the features of PyTorch. YOLOv5 makes use of a combination of CSPDarknet backbone, PANet neck and YOLOv3 head instead of the Darknet in YOLOv4. The activation function in the last detection layer is a nonlinear activation function (e.g., sigmoid function) which is broadly applied to deep learning, rather than the mish function in YOLOv4. In addition, YOLOv5 also uses auto-learning bounding box anchors to fine-tuning and optimize anchor selection.

We choose YOLOv5 as our algorithm for road sign recognition from satellite images. First, YOLOv5 incorporate cross stage partial network (CSPNet) [29] into Darknet and created CSPDarknet as its backbone. CSPNet solves the problem of repeated gradient information in large-scale trunk and integrates gradient changes into feature map, thus reduces model parameters and floating-point operations per second, which not only ensures the speed and accuracy of reasoning but also reduces the size of the model. In the task of acquiring road sign images for satellite radar sensors, visual object detection speed and accuracy are essential, the compact model is also conducive to its reasoning efficiency on resource-poor edge equipment.

Second, YOLOv5 applies a path aggregation network (PANet) [30] as its neck to boost information flow. PANet adopts a new feature pyramid network (FPN) structure with an enhanced bottom-up path, which improves the propagation of low-level features. At the same time, adaptive feature pooling, which links the feature grid and all feature levels, is employed to make useful information in each feature level propagate directly to the following subnetwork. PANet improves the utilization of accurate localization signals in lower layers, which obviously enhances the location accuracy of a visual object.

Finally, the head of YOLOv5 generates 3 different resolutions of feature maps to achieve multiscale prediction, enables the model to handle small, medium, and big objects. Traffic signs come in various types and resolutions. Multiscale [31] detection ensures that the model can follow the scale changes in the process of vehicle travel and weather changes. The training objective function of our final YOLOv5-based satellite image sign recognition takes use of the improved loss function as shown in Eq. (7).

Our Results and Discussion

Our Datasets

Dataset GTSDB contains 900 images with a total of 1,206 traffic signs. There are four types of traffic signs: Mandatory, prohibit, danger, and others. As there are not many foggy scenes in GTSDB, we take advantage of FRIDA, FRIDA2, and FROSI databases. FRIDA consists of 90 images of 18 urban road scenes, meanwhile FRIDA2 has 330 composite images of 66 road scenes. They have the same viewpoint from drivers' view, with the four types of fogs (i.e., uniform fog, heterogeneous fog, cloudy fog, and cloudy heterogeneous fog) added to each sign (i.e., Give Way, Watch Out for Pedestrians, etc.) The FROSI dataset contains foggy images with visibility ranging from 50 to 400 m, including 1,620 traffic signs at various locations. With these datasets, it is possible to train our YOLOv5 model and Faster R-CNN model much comprehensively. In this paper, in our TSR experiments from the drivers' view, we combine two datasets for training and testing. Among them, 60.00% of images were used for training, 20.00% were employed for verification, and 20.00% were utilized for testing.

In the experiment of identifying road signs based on satellite images, we were the use the dataset that we have created ourselves. There are 1,000 images captured from Google Earth, and each image is manually labelled. In this dataset, we mainly include the traffic signs like straight, right, left, give way, stop, crosswalk, keep clear, etc. This is shown in Fig. 6. In the dataset, each sample of identifiers is not uniform, straight-line sign is the most and the bicycle lane is the least. Again, we adopt 60.00% for training, 20.00% for validating, and 20.00% for model testing (Fig. 7).

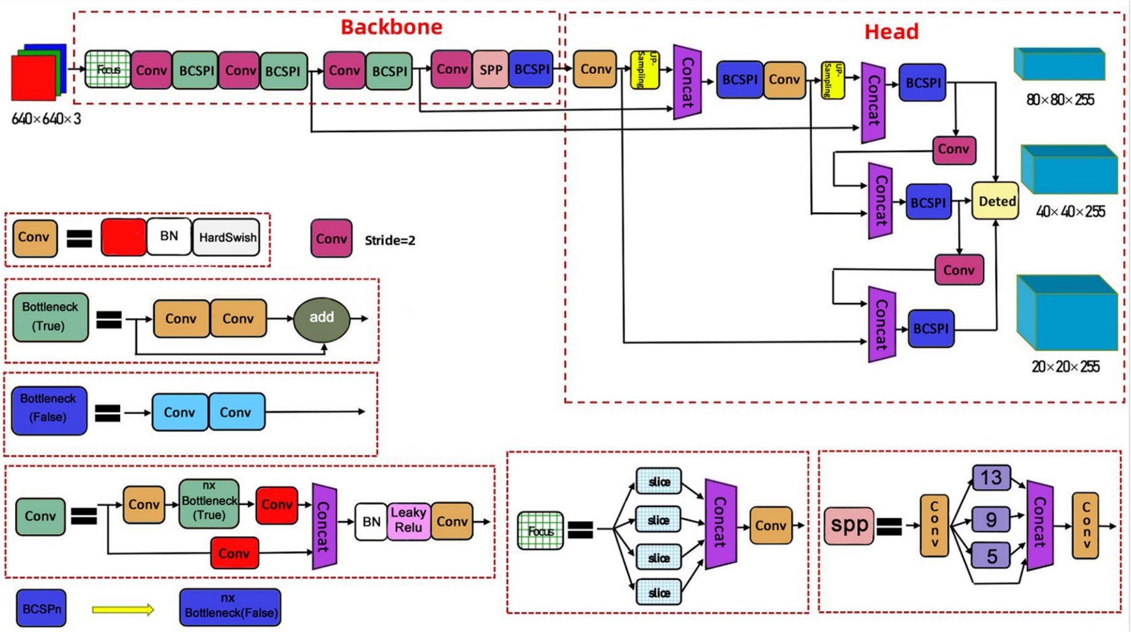


Fig. 6 The network architecture of YOLOv5

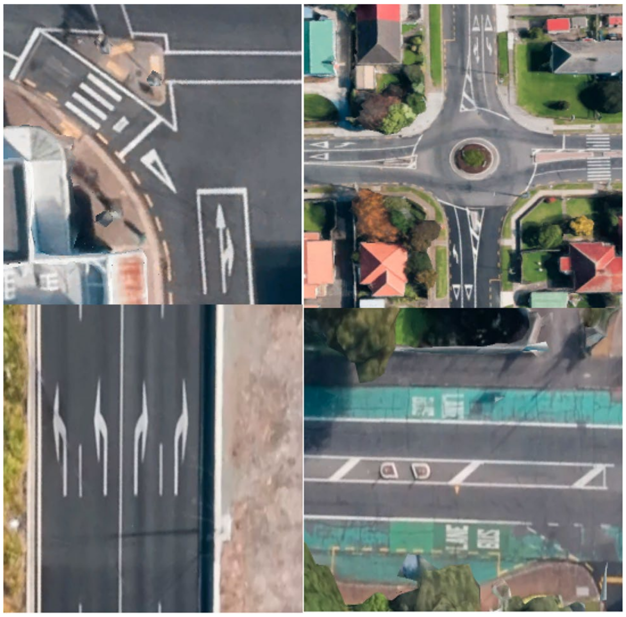


Fig.7 The satellite images of traffic signs on the roads

Evaluation Index

After training the model, it is necessary to evaluate its results. Accuracy is our most useful evaluation index, it is easy to understand, that is, the number of samples to be matched divided by the number of all samples. Generally, the higher the accuracy, the better the classifier. At the same time, we also have taken mAP and PR curves to evaluate the model. Because Precision considers the values of TP and FP in the PR curve, the precision-recall (PR) curve is more accurate than the receiver operating characteristic (ROC) curve under unbalanced data.

In this article, the evaluation index for TSR is measured by mean average precision (mAP), which is employed in the field of visual object detection. The test results include four prediction categories: TP, FP, FN, TN. Precision is the rate that the positive sample predicted correctly including false alarms (FP). Recall is for the primary positive samples, which indicates how many of the positive samples are predicted correctly including correctly rejected (FN). Therefore, the precision rate and recall rate are calculated in Eq. (8) and Eq. (9):

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{9}$$

Comparison and Analysis of Two Defogging Models

In this section, we analyse and compare the defogging results by using the dark channel algorithm and the guided image filtering method. Figure 8 shows the output of each defogging algorithm.

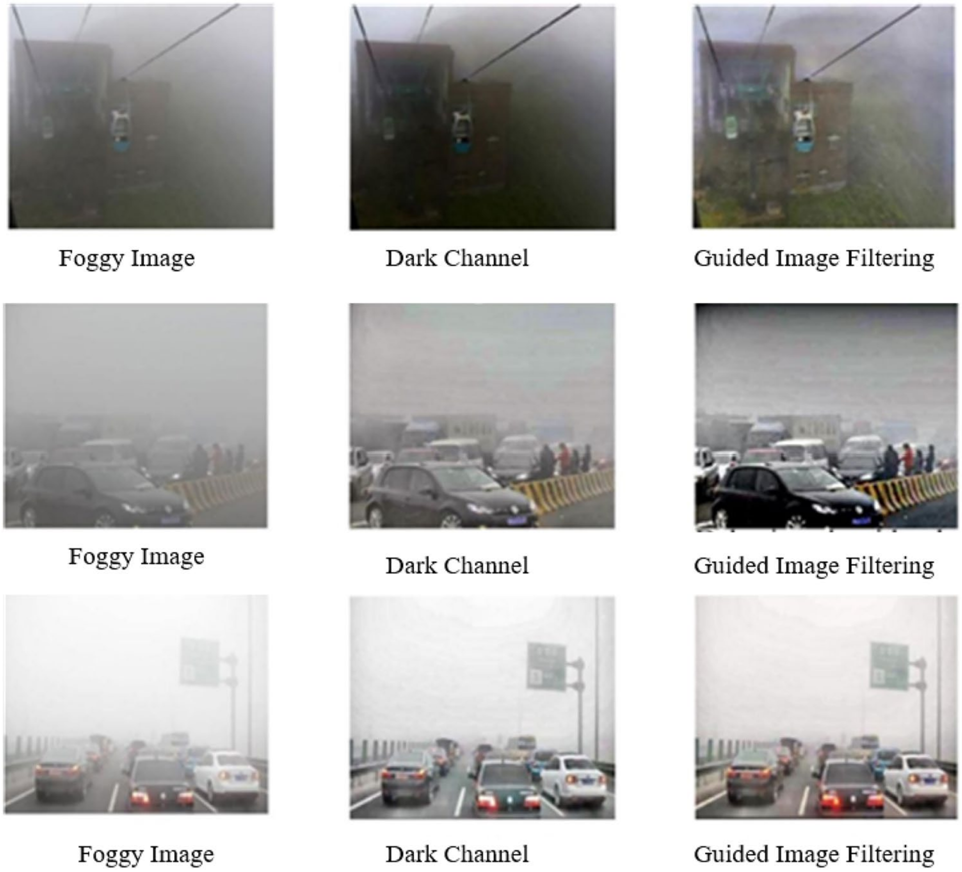

Foggy Image
Dark Channel
Guided Image Filtering
Foggy Image
Dark Channel
Guided Image Filtering
Foggy Image
Dark Channel
Guided Image Filtering

Fig.8 The results of defogging methods for various scenes

We see from the results that the defogging algorithm based on guided image filtering is much more robust and has a more stable defogging effect for multiple scenes. Guided image filtering obtains a better dehazing result, and the image colour is less distorted or darkened. At the same time, it plays a pivotal role in colour enhancement.

The Impact of Data Set Division on Experimental Results

In this experiment, to find the most suitable way to divide the data set, we split the dataset into three proportions. The ratios between training set, validation set and test sets are 4:3:3, 6:2:2, and 8:1:1. In this section, we find a suitable dataset division ratio for our experiment based on the error rate. Before calculating the error rate, we need to understand bias, variance, and noise.Bias and variance describe the gap between the model we have trained and the real model from two aspects. Bias is the error between the output result of the model based on the samples and the ground truths, which is the accuracy of the model. Variance is the error between each output result of the model and the expected value of the model output, which is the stability of the model. The error rate is obtained by adding the values of bias, variance and noises. The calculation is shown in Eq. (10).

$$\text{Error} = \mathbb{E}_D[\left(f(x;D) - \overline{f}(x)\right)^2] + (\overline{f}(x) - y)^2 + \mathbb{E}_D[(y_{(D-y)})^2] \tag{10}$$

where x is the test sample, D is the data set, y is the true mark of the test sample, $f(x)$ is the model trained with the training set D, and $f(x;D)$ is the predicted value of x for the $f(x)$ trained with the training set D, $\overline{f}(x)$ is the predicted value of model $f(x)$ for x. We first calculate the error rates related to the ratio 4:3:3, the error rate of the training set is 5.70%, and the error rate of the validation set is 8.10%. Secondly, we calculate the error rates of the ratio 6:2:2, the error rate of the training set is 3.10%, and the error rate of the validation set is 4.30%. Finally, we calculated that when the dataset is divided into 8:1:1, the error rate of the training set is 1.00%, and the error of the validation set is 7.40%.

In the experiment, we see that if the data set is divided into 4:3:3, the error rates of the verification set and the test set are relatively high, which indicates that the training is not enough. More training samples are needed. We split the dataset into 6:2:2, and then the test results are ideal. The error rates of the training set and the validation set are reduced, and the difference between the two is kept at 1.20%,

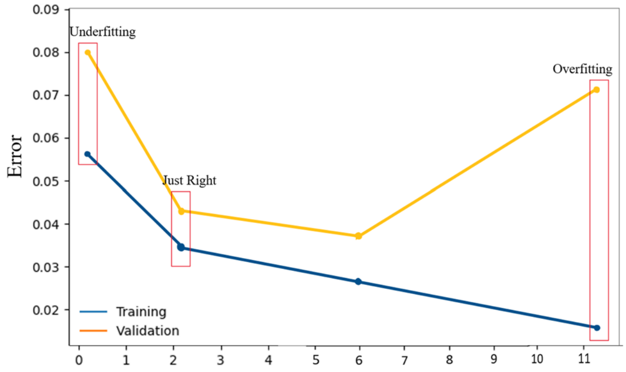


Fig. 9 The error rate curve of the training set and validation set

Table 2 The performance of different networks based on our results

Networks	Recall (%)	Precision (%)	mAP (%)	fps
VGGNet	88.2	89.1	90.2	16
GoogLeNet	88.7	93.2	95.3	17
ResNet	92.8	91.2	95.2	16

Table 3 Contrast experiment with the basic Faster R-CNN net

Methods	Recall (%)	Precision (%)	mAP (%)
Faster R-CNN	90.60	91.30	80.30
Our method	92.60	93.40	95.30

which is a good result. Finally, we segment the dataset into the ratio 8:1:1. From the experimental results, we found that though the error rate of the training set has dropped to 1.00%, the error rate of the validation set has risen to 7.40%. This is a manifestation of overfitting, such a model does not have generalization, as shown in Fig. 9. Hence, in this article, we divide the data into 60.00% for training, 20% for validating and 20.00% for testing.

TSR from Drivers' View

Our Results of Improved Faster R-CNN

In our experiments, we test various backbone networks. The performance of the network depends on the ability of the network. Therefore, the part of feature extraction that directly affects network performance requires much effort. In this paper, we offer classic networks as the feature extraction network of Faster R-CNN to compare the impact of different networks on classification performance. Table 2 shows the experimental results of different networks. We see that different backbone networks have positive results. GoogLeNet and ResNet both have an improvement of 5.10% compared to VGG net, meanwhile the running time of GoogLeNet is similar to that of VGG. Therefore, considering mAP and running time, Faster R-CNN as an object detector, GoogLeNet is employed as the backbone network.

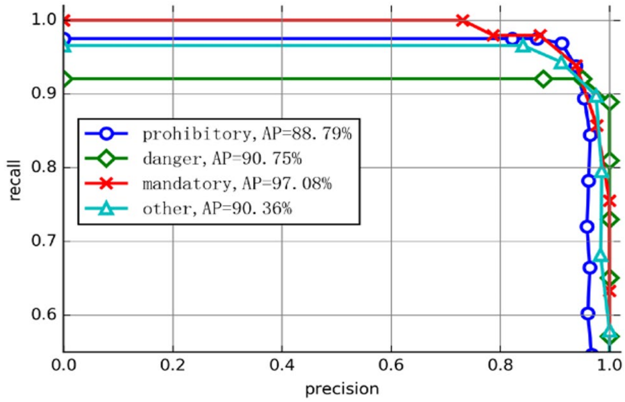


Fig. 10 PR curves of our experimental results

Table 4 The comparison of our experimental results

Methods	Recall (%)	Accuracy (%)
YOLOv5	96.56	94.30
Improved YOLOv5	97.55	95.63

Next, we tackle the image with guided filtering and input the augmented images into the designed network for classifying the traffic signs. We compare the basic Faster R-CNN network. Table 3 shows the specific performance of our proposed method based on the given dataset.

In Table 2, we compare the accuracy, recall and precision rates of the three nets. We see that under the current scale of data training, GoogLeNet is better than the VGG net in recall and accuracy, but the running time will be relatively slower than the VGG net. Compared with ResNet, our recall rate is relatively low, other metrics are rather better. However, it costs a little bit longer time.

In Table 3, the recall and accuracy rates of Faster R-CNN are relatively high. The reason is that there are a large number of traffic signs in reality. Accordingly, we took use of guided image filtering for processing the images. The feature fusion method based on GoogLeNet is proposed in this paper for model training. Although the recall of target detection has not been improved too much, the accuracy has increased by 15.00%, which is explained that by adding difficult negative samples, the capability of the net has been increased a lot owning to the image enhancement. Figure 10 shows the PR curves for four different classifiers. In complicated scenes, the general model usually cannot detect the traffic signs well.

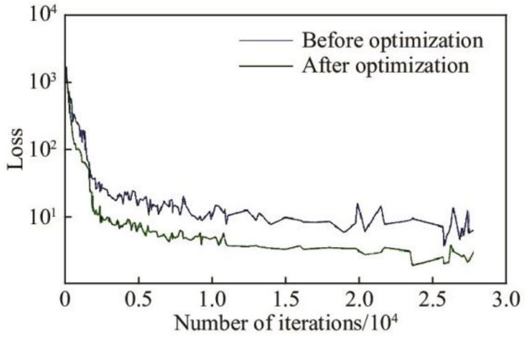


Fig.11 The loss curve with the improved deep learning models

The Result of Improved YOLOv5

Whilst training and testing YOLOv5 model, the same dataset was employed, the dataset was split in the same way, 60.00% for training, 20.00% for validating, and 20.00% for testing. In this paper, we modify YOLOv5 framework as the basis of the TSR net and train two nets separately. One of them is the standard YOLOv5 net, which is used as a comparison method. The test results of the improved YOLOv5 algorithm and the original YOLOv5 algorithm are shown in Table 4. The loss curve is shown in Fig. 11.

The Comparison of YOLOv5 and Faster R-CNN

These models with the same dataset were trained and evaluated based on a computer equipped with a Core i7-8th CPU, 16 GB of RAM, and NVIDIA RTX2060 GPU. Firstly, we compare the training time of the two. Faster R-CNN training took 14 h, YOLOv5 training spent 11 h because YOLOv5 has a smaller network size than Faster R-CNN. Secondly, we compare the recognition speed of the two methods. The detection speed of Faster R-CNN is 17 fps, and the recognition speed of YOLOv5 is 60 fps. YOLOv5 is much suitable for TSR in real time. Finally, Fig. 12a and b show the TSR results of the two nets by using the FRIDA dataset.

We also compare the recognition results of Faster R-CNN and YOLOv5 in real life. Figure 13 shows our TSR under sunny weather, Figs. 14 displays the recognition results of the two methods in foggy weather. In Fig. 13, we see the TSR results, which show that Faster R-CNN is often missed and incorrectly detected if the traffic signs are far from the camera. In contrast, YOLOv5 has higher recognition accuracy and speed when recognizing small objects or objects that move faster. Figure 14 shows the recognition result based on foggy images, which is roughly similar to the recognition result based on sunny-day images. Faster R-CNN is prone to solving the problems of low object detection rate and slow object recognition speed whilst recognising small and fast-moving objects.

The video for our tests is composed of 2,590 frames. YOLOv5 takes 9.00×10^{-3} s to cope with each frame. Faster R-CNN spends 21.00 s to deal with each frame, which takes a much longer time than YOLOv5. Under the same accuracy rate, YOLOv5 has a faster recognition speed. Because TSR is often used for real-time object detection and recognition with high requirements of computing speed, YOLOv5 is much suitable for TSR.

Guided Image Filtering

In this section, we use YOLOv5 as the basic framework to compare the recognition results with and without dehazing. With the dehazing operation, more traffic signs have been recognized. In Fig. 15a, there is a traffic sign that has been recognized after the dehazing operation.

TSR from Satellite Imagery

To further expand traffic sign recognition, we take use of the improved YOLOv5 to detect traffic signs from another view angle of satellite images. The hyperparameters of the YOLOv5 model are: Batch size and mini-batch size are 16 and 4, respectively. The momentum and weight decays are 0.90 and 0.50×10^{-3}; the initial learning rate is 0.10×10^{-2}, the epoch is 30.

Figure 16 shows multiple metrics as the number of iterations increases, where bounding box regression decreases as the iteration increases at this point, mAP values drop as the iteration raises. It shows that the detection result of the proposed net in this paper is getting much better with the growth of iteration times. The precision and recall rates are also boosted with more iterations of network parameters, this indicates that the number of positive samples also increases with the increase of the number of iterations. In general, the improved YOLOv5 model in this paper is better for the detection of road signs based on satellite images as the number of iterations increases.

Figure 17 shows PR (i.e., precision-recall) curve of our test results in this experiment, with precision rate as *y*-axis and recall rate as *x*-axis. We see that the closer the drawn PR curve is to the upper right, which proves that the YOLOv5 method has super effectiveness in road sign recognition based on the satellite image. Therefore, road sign recognition based on the satellite image is a promising prospect.

Figure 18 shows TSR results under various iteration times for satellite images. We see that the more iterations, the better the recognition outcomes. Figure 19 displays the final result with the satellite images.

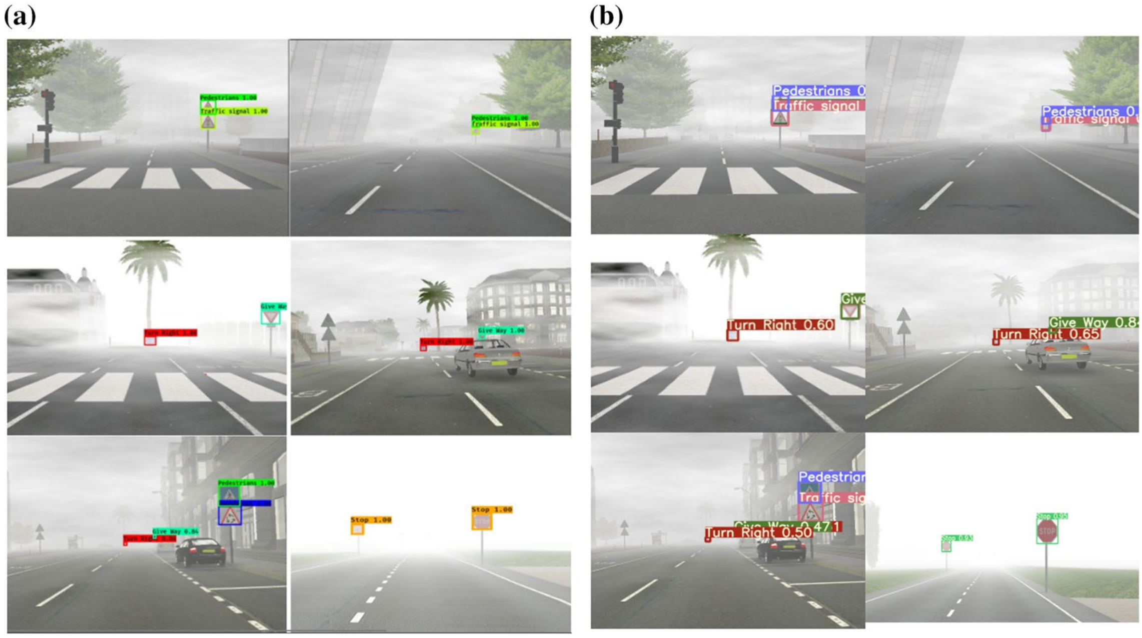
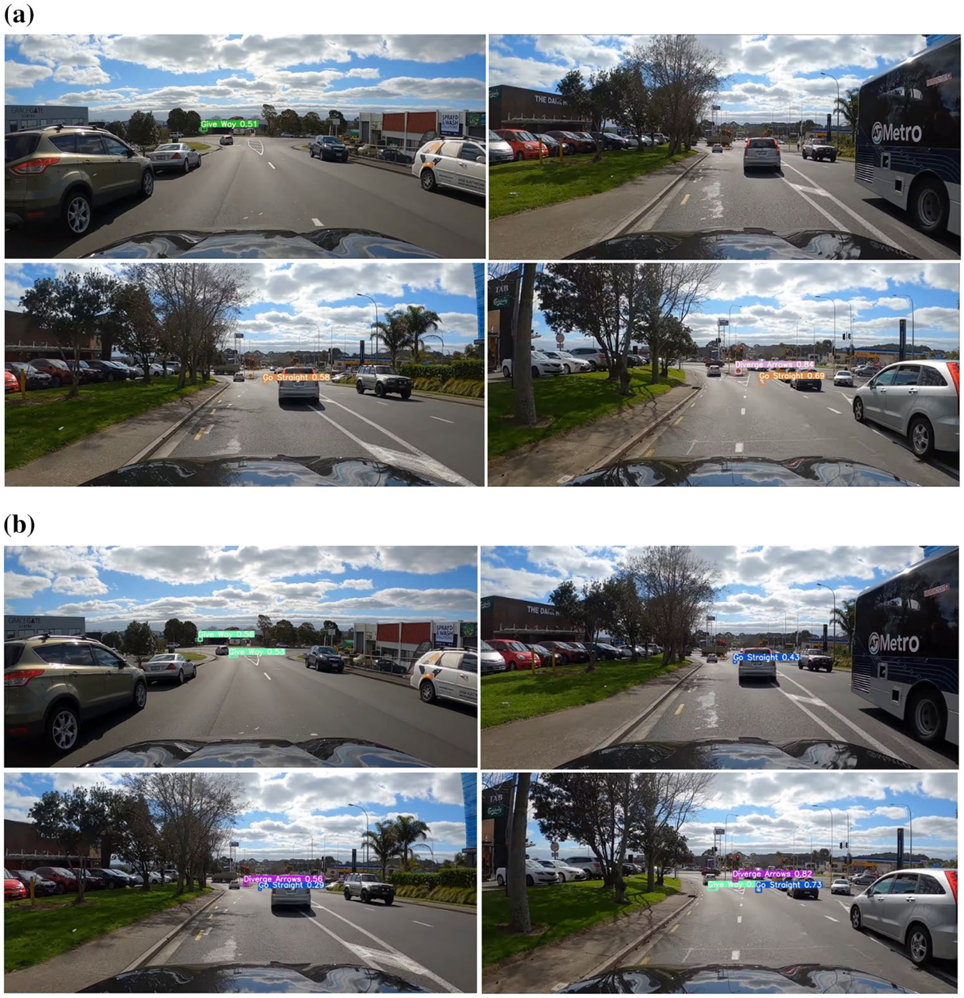


Fig. 12 The result of recognition on FRIDA dataset with YOLOv5 (a) and Faster R-CNN (b)

Fig. 13 The TSR results on sunny days. **a** Faster R-CNN, **b** YOLOv5



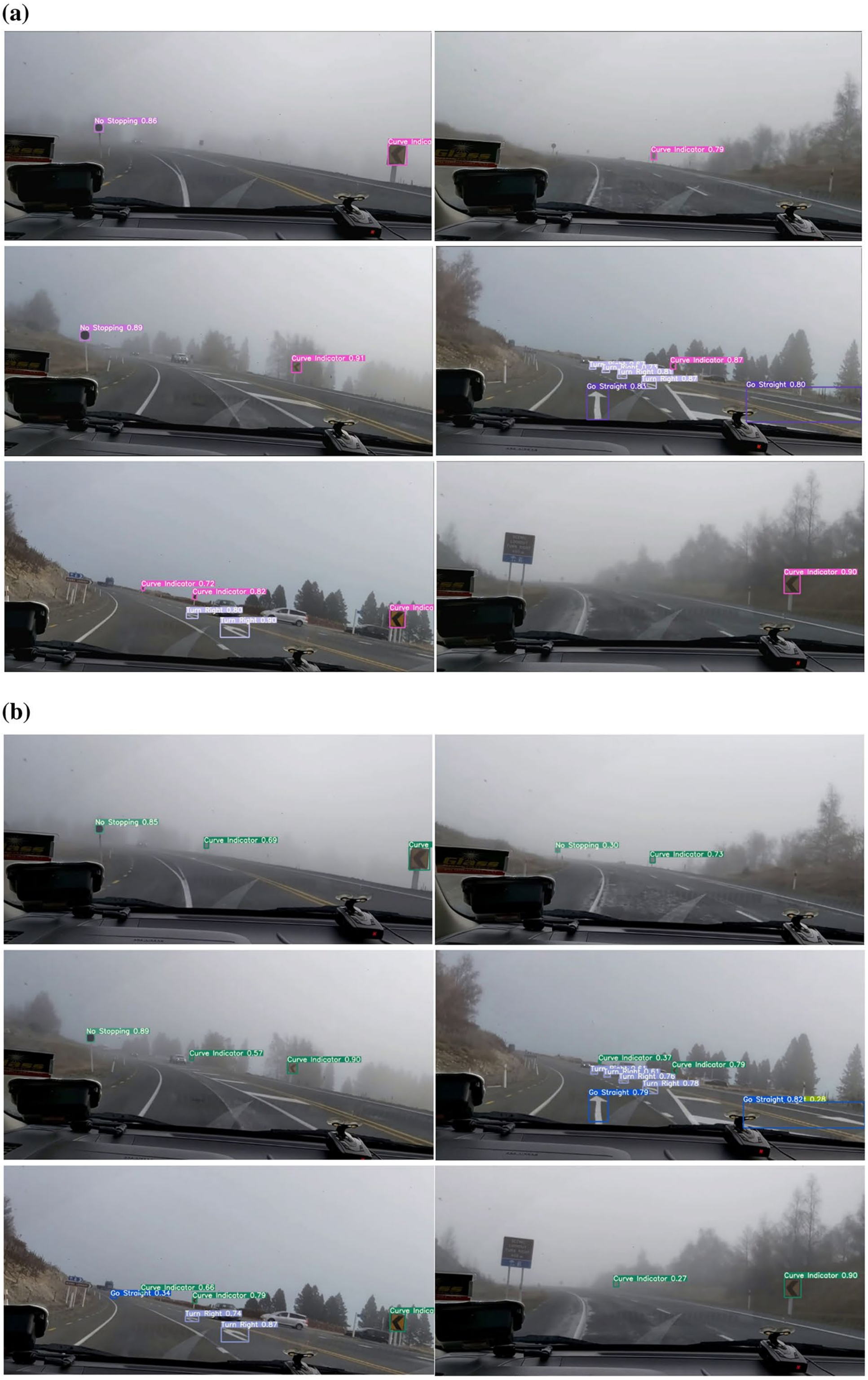
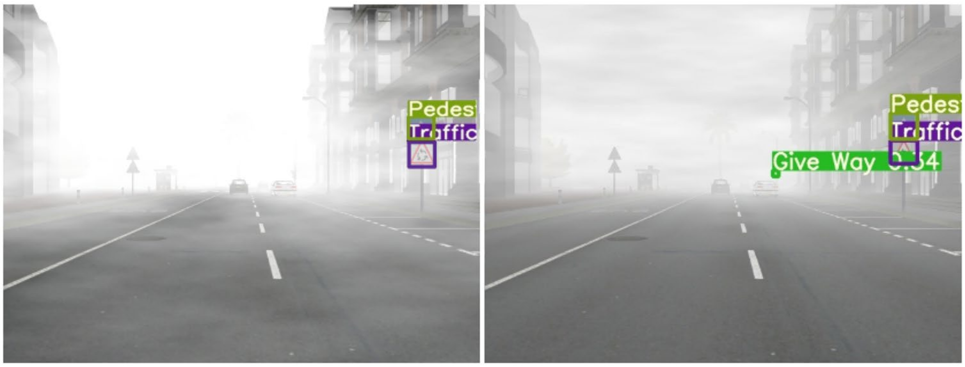


Fig. 14 TSR results based on foggy images. **a** R-CNN, **b** YOLOv5



(a) YOLOv5 without dehazing **(b)** YOLOv5 with image guided filtering

Fig. 15 TSR results with and without dehazing operation

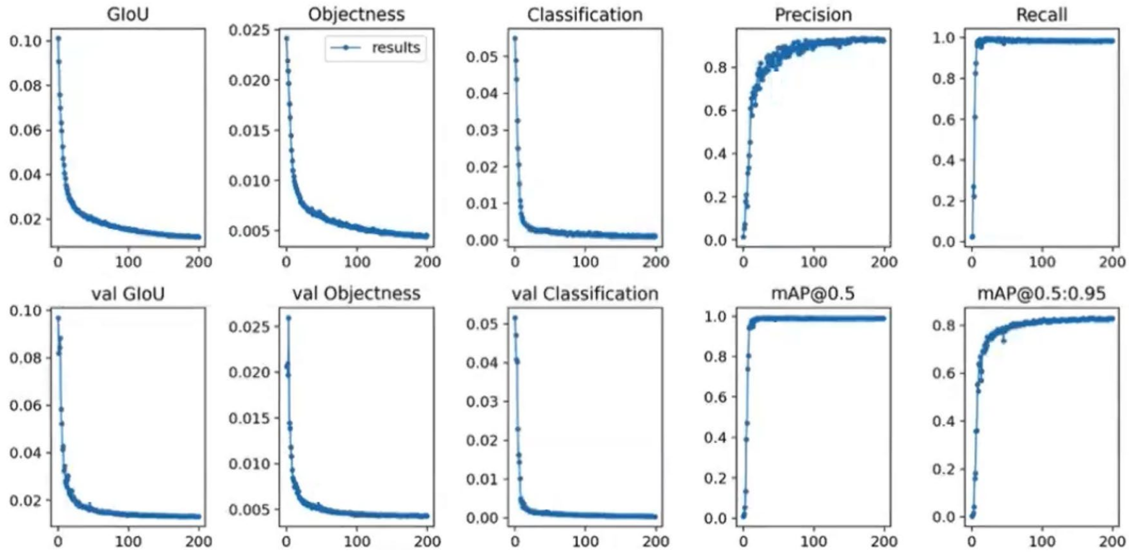


Fig.16 The changes with various metrics

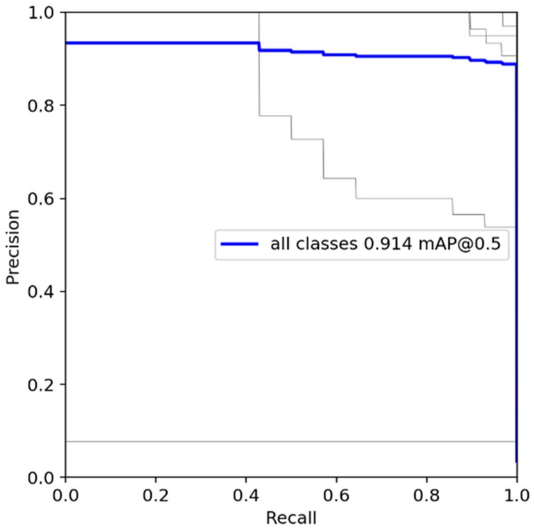


Fig.17 The PR curves for TSR based on satellite images

Conclusion

Under adverse weather conditions, the TSR accuracy is not very high. In this article, we deeply investigated Faster R-CNN and improved YOLOv5 algorithm for TSR, from the perspective of the drivers' view and satellite imagery. We compare the results of TSR recognition with multiple nets. If the overall framework of the experiments is the same, we chose the excellent network as our base net.

We have effectively employed multiresolution feature maps through cross-layer connections to build up the feature maps of traffic sign objects with multiple scales. We make use of guided image filtering to eliminate the noises from the given images, and further improve the accuracy of our experiments.

There are two aspects to our future work. One is to collect more traffic signs as samples under complicated conditions

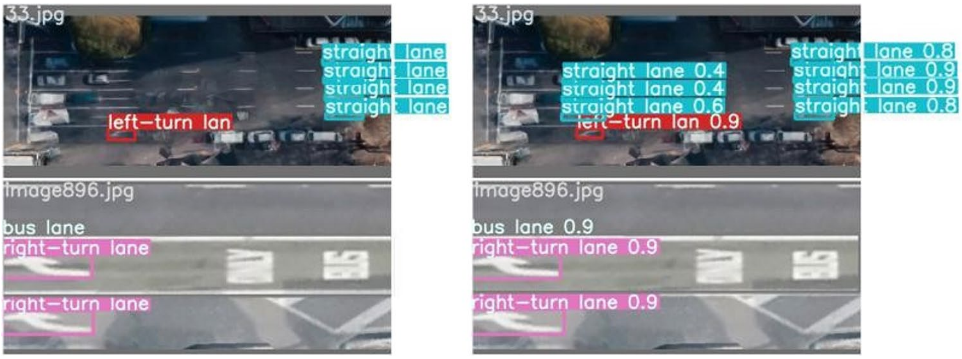


Fig. 18 TSR results with various iteration times

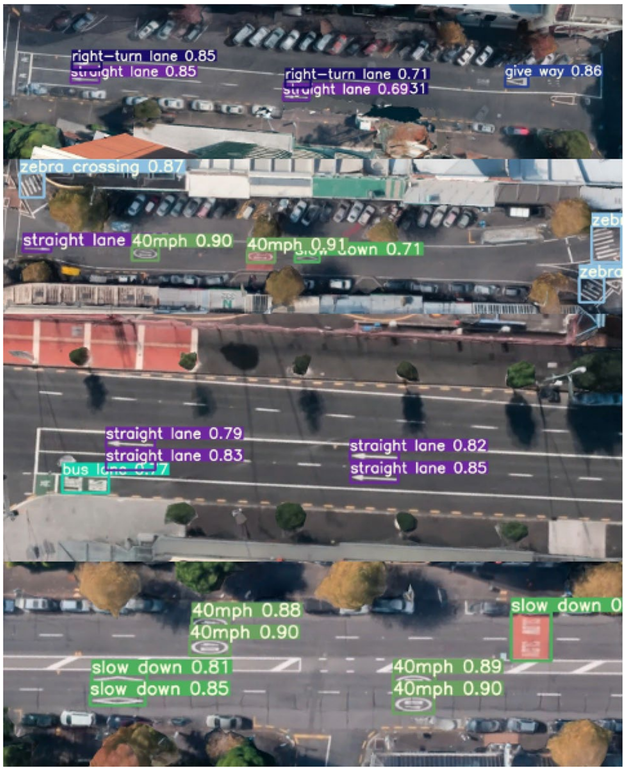


Fig. 19 TSR results with satellite images

to form our own dataset. The other is to further optimize the method to form an end-to-end TSR framework [32, 33, 36, 38, 39].

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

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C. Detection of traffic signs in real-world images: the German traffic sign detection benchmark. Int Joint Confer Neural Netw. 2013. https://doi.org/10.1109/IJCNN.2013.6706807
2. Yang Y, Luo H, Xu H, Wu F. Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst. 2016; 17(7):2022–31.
3. Berkaya SK, Gunduz H, Ozsen O, Akinlar C, Gunal S. On circular traffic sign detection and recognition. Expert Syst Appl. 2016; 48:67–75.
4. Jie Y, Xiaomin C, Pengfei G, Zhonglong X. A new traffic light detection and recognition algorithm for electronic travel aid. Int Confer Intell Control Inform Process. 2013. https://doi.org/10.1109/ICICIP.2013.6568153.
5. Jin J, Fu K, Zhang C. Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Trans Intell Transp Syst. 2014;15(5):1991–2000.
6. Priese L, Klieber J, Lakmann R, Rehrmann V, Schian R. New results on traffic sign recognition. IEEE Intell Vehicles Symp. https://doi.org/10.1109/IVS.1994.639514.
7. Sun L, Chen J, Xie K, Gu T. Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition. Int J Speech Technol. 2018; 21(4):1–10.

8. Ren Y, Yang J, Zhang Q, Guo Z. Multi-feature fusion with convolutional neural network for ship classification in optical images. Appl Sci. 2019; 9(20):4209.
9. Ruta A, Li Y, Liu X. Detection, tracking and recognition of traffic signs from video input. Intell Transp Syst. 2008; 55–60. https://doi.org/10.1109/ITSC.2008.4732535.
10. Blancard M. Road sign recognition: a study of vision-based decision making for road environment recognition. Vision-Based Vehicle Guidance, 1992; 162–172. https://doi.org/10.1007/978-1-4612-2778-6_7.
11. Girshick R, Donahue J, Darrell T. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE CVPR. 2014. https://doi.org/10.1109/CVPR.2014.81.
12. Uijlings R, Sande A, Gevers T, Smeulders M. Selective search for object recognition. Int J Comput Vision. 2013; 104(2):154–71.
13. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell. 2014; 37(9):1904–16.
14. Girshick R. Fast R-CNN. IEEE International Conference on Computer Vision. 2015. https://doi.org/10.1109/ICCV.2015.169.
15. Müller J, Dietmayer K. Detecting traffic lights by single shot detection. IEEE International Conference on Intelligent Transportation Systems (ITSC). 2018. https://doi.org/10.1109/ITSC.2018.8569683.
16. Douville P. Real-time classification of traffic signs. Real-Time Imaging. 2000; 6(3):185–93.
17. Barnes N, Zelinsky A. Real-time speed sign detection using the radial symmetry detector. IEEE Trans Intell Transp Syst. 2016; 9(2):322–32.
18. Huang X, Zhang L. Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines. Int J Remote Sens. 2009; 30(8):1977–87.
19. Line E. Classification-based vehicle detection in high-resolution satellite images. J Photogramm Remote Sens. 2009; 64(1):65–72.
20. Leitloff J, Hinz S, Stilla U. Vehicle detection in very high resolution satellite images of city areas. IEEE Trans Geosci Remote Sens. 2010; 48(7):2795–806.
21. Audebert N, Saux B, Sébastien L. Semantic segmentation of earth observation data using multimodal and multiscale deep networks. Asian Confer Comput Vis. 2016; 180–96.
22. Volpi M, Tuia D. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. IEEE Trans Geosci Remote Sens. 2017; 55(2):881–93.
23. Sherrah J. Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv:1606.02585. 2016.
24. Cheng G, Zhou P, Han J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans Geosci Remote Sens. 2016; 54(12):7405–15.
25. Illingworth J, Kittler J. A survey of the Hough transform. Comput Vis Graph Image Process. 1988; 43(2):280–280.
26. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, 2015. https://doi.org/10.48550/arXiv.1409.1556.
27. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. IEEE CVPE. 2015.
28. Sung K. Learning and example selection for object and pattern detection. MIT PhD Thesis. 1996.
29. Wang Y, Liao M, Wu H, et al. CSPNet: a new backbone that can enhance learning capability of CNN. IEEE/CVF CVPR Workshops. 2020.
30. Wang K, Liew H, Zou Y, et al. PANet: few-shot image semantic segmentation with prototype alignment. IEEE ICCV. 2019; 9197–206.
31. Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767. 2018.
32. Xing J, Yan W. Traffic sign recognition using guided image filtering. Int Symp Geometry Visi. Springer CCIS 1386. 2021; pp. 85–99.
33. Yan W. Computational methods for deep learning: Theoretic, practice and applications: Springer, 2021.
34. Bayoudh K, Hamdaoui F, Mtibaa A. Transfer learning-based hybrid 2D–3D CNN for traffic sign recognition and semantic road detection applied in advanced driver assistance systems. Appl Intell. 2021; 51(1):124–42.
35. Bi Z, Yu L, Gao H, Zhou P, Yao H. Improved VGG model-based efficient traffic sign recognition for safe driving in 5G scenarios. Int J Mach Learn Cybern. 2020; 1–12.
36. Yang X, Liu W, Zhang S, Liu W, Tao D. Targeted attention attack on deep learning models in road sign recognition. IEEE Internet Things J. 2020;8(6):4980–90.
37. Jin Y, Fu Y, Wang W, Guo J, Ren C, Xiang X. Multi-feature fusion and enhancement single shot detector for traffic sign recognition. IEEE Access. 2020;8:38931–40.
38. Yan W. Introduction to intelligent surveillance—surveillance data capture, transmission, and analytics. 3rd ed. New York: Springer; 2019.
39. Xing J. Traffic sign recognition from digital images by using deep learning. Masters Thesis, Auckland University of Technology, New Zealand. 2021.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.