Small Traffic Sign Detection and Recognition in High-Resolution Images

Lei You[1], Yu Ke[1], Hongpeng Wang[1](✉), Wenhu You[2], Bo Wu[3],
and Xinghao Song[1]

[1] HIT Campus of University Town of Shenzhen, Shenzhen 518055, China
wanghp@hit.edu.cn
[2] No. 92 West Da Jie, Nangang District, Harbin 150001, China
[3] No. 2190, Liuxian Street Nanshan, District, Shenzhen 518055, China

Abstract. Traffic sign detection and recognition is a research hotspot in the computer vision and intelligent transportation systems fields. It plays an important role in driver-assistance systems and driverless operation. Detecting signs, especially small ones, remains challenging under a variety of road traffic conditions. In this manuscript, we propose an end-to-end deep learning model for detecting and recognizing traffic signs in high-resolution images. The model consists of basic feature extraction and multi-task learning. In the first part, a network with fewer parameters is proposed, and an effective feature fusion strategy is adopted to gain a more distinct representation. In the second part, multi-task learning is conducted on different hierarchical layers by considering the difference between the detection and classification tasks. The detection results on two newly published traffic sign benchmarks (Tsinghua-Tencent 100K and CTSD) demonstrate the robustness and superiority of our model.

Keywords: Traffic Sign Detection · Traffic Sign Recognition · Small traffic sign detection · End-to-end detection and recognition

1 Introduction

Traffic sign detection (TSD) and traffic sign recognition (TSR) are important components of autonomous driving and assisted driving, which are designed to help drivers avoid traffic accidents. Traffic sign detection aims to find areas that contain a traffic sign. Traffic sign recognition aims to determine the specific categories of these areas. The automatic detection and recognition of traffic signs can improve driving safety and comfort. However, effective traffic sign detection is not easy to achieve when faced with complex natural scenes. On the other hand, as the speed increases, distant traffic signs, which are relatively small and very difficult to detect, need to be detected in advance.

Many approaches have been proposed to address TSD and TSR; a good review [1] shows the efforts that have been made in recent years. Convolutional neural network (CNN) methods [2–5] have achieved perfect or near-perfect results on the German Traffic Sign Detection Benchmark (GTSDB) [6] and German Traffic Sign Recognition Benchmark (GTSRB) [7]. The reason for this is that CNNs generate more discriminative

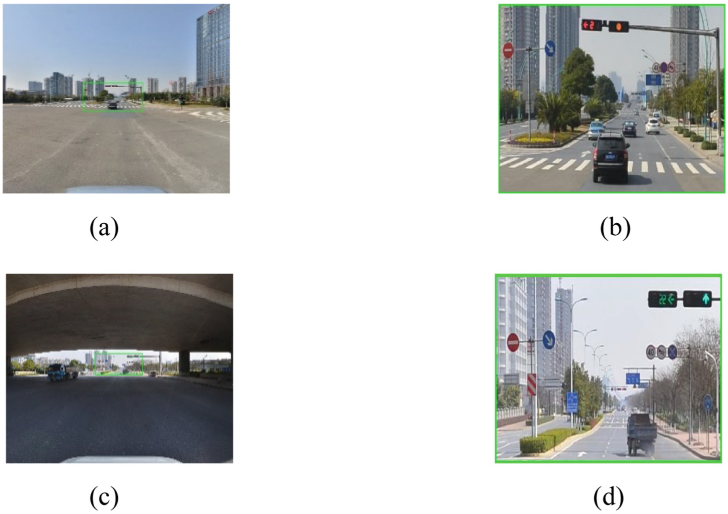
R. Xu et al. (Eds.): ICCC 2019, LNCS 11518, pp. 37–53, 2019.
https://doi.org/10.1007/978-3-030-23407-2_4

features than traditional methods, such as color segmentation [8–10] and shape detection [11, 12]. However, the number of datasets for TSD and TSR is relatively small and not representative of a real environment. For example, the GTSDB contains only 900 images, while most of the images in the GTSRB contain traffic signs; thus, we need only to perform image classification. Similar to other computer vision tasks, deep learning algorithms [13–15] are also used to detect and recognize traffic signs.

At present, the proportion of traffic signs in most benchmarks is relatively large, which indicates that when a vehicle is close to a traffic sign, the remaining driver reaction time will be very short. In this work, we focus on relatively small traffic signs in the distance. We propose a multi-task deep learning framework to simultaneously detect and recognize traffic signs in an image. Compared with previous work [16], our method performs better in terms of recall and accuracy. As illustrated in Fig. 3, our model consists of feature extraction and multi-task learning. In the first part, the model gains a more distinct representation via feature fusion from multiple layers. In the second part, multi-task learning based on detection and classification, which includes three branches: bbox branch, type branch and pixel branch, is carried out.

We perform experiments on the Tsinghua-Tencent 100K (TT100K) [16] and Chinese Traffic Sign Dataset (CTSD) [17] to illustrate the effectiveness and efficiency of our proposed method. The size of many traffic signs in TT100K is approximately 20 * 20 pixels, and the signs occupy less than 1/10000 of the area of their respective images. Image samples from the TT100K benchmark are shown in Fig. 1. The sizes of traffic signs in the CTSD are relatively larger; samples of the CTSD are shown in Fig. 2.

Our main contributions are as follows. First, we propose a feature extraction network with fewer parameters that outperforms the work in [16]. Second, we add a



(a)

(b)

(c)

(d)

Fig. 1. TT100K image samples. (a) (c) Original images: the green rectangle represents the area that contains small traffic signs. (b) (d) Image patches that contain the small traffic signs corresponding to the original images

"bottleneck" layer, which is similar to ResNet [18], to fuse the feature maps from different layers, which improves the recall of the TSD. Third, multi-task learning on different hierarchical feature maps is conducted to further improve the result, especially for small traffic signs. Finally, the proposed method is shown to achieve state-of-the-art results on TT100K and comparable results on the CTSD.

The remainder of this paper is organized as follows. In Sect. 2, we review previous work. In Sect. 3, we describe the network structure of the model. Detailed experiments are introduced in Sect. 4, and we conclude this paper in Sect. 5.

(a)

(b)

Fig. 2. CTSD image samples. The benchmark includes images at two different resolutions.

2 Related Work

2.1 Traffic Sign Detection

Traffic Sign Detection Traffic sign detection aims to segment the regions of an image that may contain traffic signs. Three main approaches to TSD exist: the color-based, shape-based and sliding-window methods. The color-based method is used to identify specific regions in various color spaces. Since the original RGB color space is sensitive to changing light, other color spaces, such as the HSI color space [8, 19] and Lab color space [9], are adopted to improve performance. For example, the authors of [20, 21] adopt color enhancement in the three channels of the RGB color space to filter regions that do not contain traffic signs. The author of [17] uses color probability maps and maximally stable extremal regions to generate candidates. The shape-based method detects specific circular, triangular and octagonal regions. The Hough transform [12], fast radial symmetry [11], log-polar transformation [22] and distance to the border [23] are commonly used to detect the shape information of traffic signs. For example, [24] proposes a saliency model for identifying sign-specific color, shape and spatial location information to facilitate detection. The sliding-window method goes through all possible positions of the traffic signs. The shortcoming of these methods is that they are usually time-consuming. For example, a coarse-to-fine algorithm for TSD is presented in [25]. The algorithm roughly identifies all candidates by using a small sliding window and relocates these candidates to a larger window for further verification. Although this method achieves state-of-the-art detection results on the GTSDB, it is extremely time-consuming when the images have high resolution.

2.2 Traffic Sign Recognition

Traffic sign recognition aims to classify candidate regions into subclasses. The traditional methods based on handcrafted features, the HOG [8, 26, 27] and the LBP [28] are used to train classifiers such as the SVM [23, 28, 29], k-NN [19], ELM [30] and sparse representation [31, 32]. Compared with handcrafted features, the features learned by neural networks from a large amount of data are more discriminative. Most methods that achieve state-of-the-art results on the GTSRB utilize neural networks for classification. For example, a committee of CNNs is used to perform classification in [3]. Several CNNs trained on differently preprocessed data are combined into a multi-column CNN to perform classification in [4]. The authors of [5] use a CNN with a hinge loss stochastic gradient descent function to perform classification and achieve a nearly perfect result that is better than human performance. Most of the methods above consider only classification and do not involve end-to-end detection and recognition.

2.3 Object Detection Based on Deep CNNs

Deep convolutional neural networks have attracted substantial attention because of their outstanding performance in object detection and classification. We classify deep CNNs into region-proposal-based methods and others. Region-proposal-based methods predict the number of candidates that contain targets and then perform classification and regression on these candidates. For example, in Fast-RCNN [33], selective search is used to generate approximately 1000 candidates. Based on Fast-RCNN, Faster-RCNN [15] integrates the region proposal network (RPN) into the entire network to accelerate detection. R-FCN [14] is a region-based method that improves the detection and recognition accuracy by generating position-sensitive maps and has achieved state-of-the-art results on PASCAL VOC 2012. Other methods that are not based on a region proposal, such as YOLO [34] and SSD [13], aim at real-time detection performance. These methods are much faster than region-based methods and have relatively lower detection and recognition accuracy.

Deep CNNs have also been used in intelligent transportation systems. For example, the authors of [35] propose a deep neural network based on Overfeat [36] to detect cars and lanes on a highway. Based on [35], the authors of [16] propose multi-task learning to simultaneously detect and recognize small traffic signs in images. A fully convolutional network (FCN) is used in [37] to generate a heat map of the traffic signs. The candidates are generated on these heat maps and are then classified by a shallow CNN.

Small object detection is an open challenge, with few methods showing good performance. In [38], the authors design a detector to detect tiny faces in an image. The contextual information and resolution information are fully used to enhance the accuracy of small face detection. A network based on SSD [13] is used to detect small traffic signs in [39]. Based on an image pyramid strategy and SOS-CNN, this method has achieved state-of-the-art results on TT100K. Perceptual generative adversarial networks (GANs) [40] are used to learn more discriminative feature representations of small traffic signs. GANs effectively detected small traffic signs on TT100K.

We propose an end-to-end learning structure to solve the problem of small traffic sign detection and recognition. The performance of the proposed method is evaluated

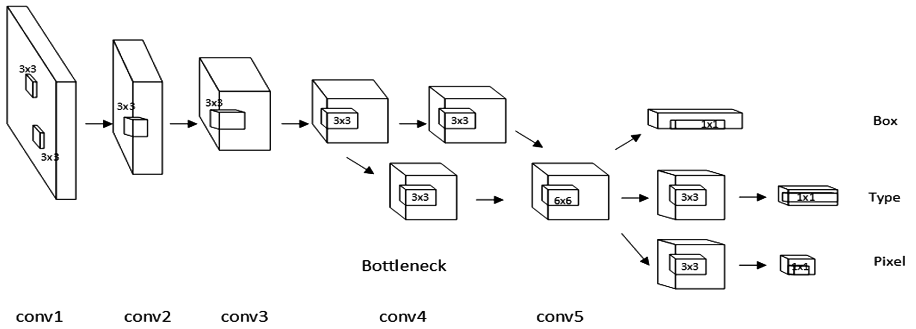


Fig. 3. The architecture of our network, which consists of four convolution blocks, a “bottleneck” block and three branches after the fifth block.

on the TT100K dataset. The CTSD is used to evaluate the generality of our model. Our work is based on the method in [16] but with several significant modifications. The implementation details and experimental results are described later.

3 Our Work

Our proposed network has an architecture similar to that of FCNs, as shown in Fig. 3, with fewer parameters and being insensitive to image size. The first half of the network is a feature extraction subnetwork similar to the VGG net [41]; a new feature fusion strategy that is similar to ResNet is also adopted. The second half of the network is a multi-task learning subnetwork that includes three branches: the bbox branch, pixel branch and type branch. For the bbox branch, each result represents the distance between the original 4 * 4 pixel region and the four sides of the predicted bounding box of the target. The pixel branch represents the probability that a certain 4 * 4 pixel region in the input image contains a target object. The type branch represents the probability of a specific class. The visualization results of the type branch and pixel branch are shown in Fig. 4. Both general traffic signs and small traffic signs can be detected on the heat maps.

3.1 Feature Extraction Network

For the first part of the network, we proposed a feature extraction network with fewer parameters, which has better performance than that in [16]. To some extent, the good performance of deep learning depends on a large number of data sets. Some commonly used benchmarks for deep learning are shown in Table 1, and the sizes of some deep learning models are shown in Table 2. Clearly, the amount of data and the size of each model are unbalanced. In general, models with larger parameters require more training data. Considering the design of the VGG network, we propose a feature extraction network with fewer parameters. The parameters of the new model reflect a reduction of 75% compared with the previous model. The sizes of the convolution kernel are mainly 3 * 3 and 1 * 1. The fifth convolution layer adopts a relatively large convolution

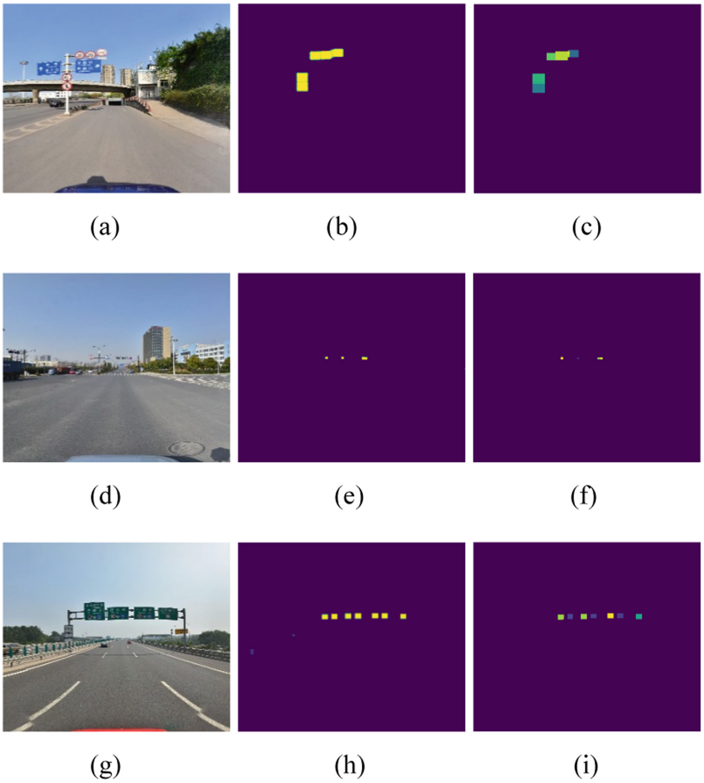


Fig. 4. The original images and heat maps of different branches. (a) (d) (g) The original testing image. (b) (e) (h) Heat maps of the pixel branch, which represents the probability that the corresponding position contains the target. (c) (f) (i) Heat maps of the type branch, which represents the probability that the corresponding position belongs to a specific class

Table 1. Open deep learning benchmarks

Benchmark	Number (k)	Labels
ILSVRC 2012	1200	1000
Open Images Dataset	9000	6000
Microsoft COCO	330	80
PASCOL VOC	22	20
TT100K	9.7	200
CTSD	1.1	48

kernel size of 6 * 6 to learn more contextual information and own larger receptive fields. These two tricks are proven to be effective for detecting small objects in [38]. The number of convolution steps of the feature extraction network is 16, and the final feature maps are twice as large as before.

Table 2. Commonly used deep learning model size

Model	Faster-RCNN	R-FCN	YOLO	SSD	Model [1]
Size (MB)	528	203	257	114	446

3.2 Feature Fusion Strategy

The sizes of traffic signs in TT100K vary widely, with most being of small size. The size distribution is shown in Fig. 5. We show the visualization results of the last two convolutional feature maps for two different sizes of traffic signs in Fig. 6. In general, we need more detailed information to detect small targets, which is not always true for large targets. Though the model can detect more small traffic signs if we use the information of convolution layer conv4, the accuracy of the classification will decrease. Considering the difference between small traffic sign detection and large traffic sign detection, we propose a feature fusion strategy that is similar to ResNet [18]. A “bottleneck” layer, which is used to fuse the feature maps of the last two convolution layers, is added to the feature extraction network. The final feature maps retain more details. In addition, our feature fusion strategy detects more traffic signs while ensuring classification accuracy.

3.3 Hierarchical Multi-task Learning

In deep learning, differences exist in the characteristic information needed by the target detection and classification tasks. In general, the larger the number of convolutional networks, the smaller the size of the convolutional feature maps. The classification task requires high-level features that are more representative. However, this is not true for the small target detection task. If the size of the feature maps is too small, it will not be conducive to the detection of small traffic signs. Considering the difference, we perform multi-task learning on different hierarchical feature maps. The detection task is based on relatively larger feature maps, while the classification task is based on higher feature maps. Specifically, the convolution step of the pixel branch and the type branch is 32, while the convolution step is 16 for the box branch. We add additional convolutional layers to the pixel branch and type branch. Based on this strategy, we further improve the detection recall rate of small traffic signs. The overall recall rate and accuracy rate on the TT100K benchmark are 94% and 91%, respectively, which are 3% and 3% higher than before.

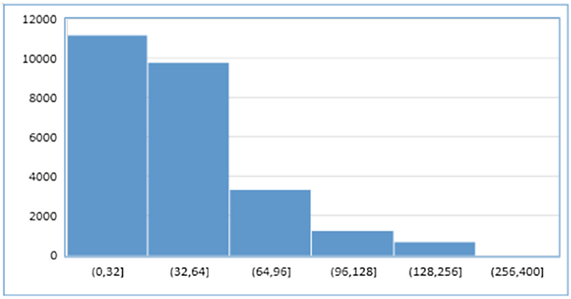


Fig. 5. The size distribution of traffic signs in TT100K.

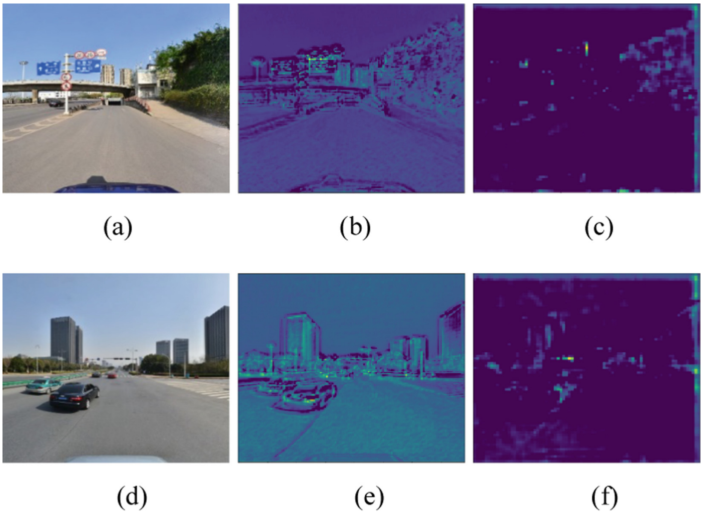


Fig. 6. The visualization results of the last two convolutional feature maps. (a) (d) The original testing image. (b) (e) The visualization results of conv4. (c) (f) The visualization results of conv5.

3.4 Training

Our experimental environment is a Linux PC with a single NVIDIA Tesla K40 GPU and 12 GB of memory. Our deep learning framework is Caffe. Images in the TT100K benchmark cannot be trained directly because of the image sizes and imbalance among traffic sign classes. To address the first problem, we randomly resize the original images within a certain range. A rectangle of 640 * 480 is utilized to segment the resized image according to the ground truth, which represents the position of the bounding box. For the second problem, we place the standard template of each traffic sign on a background of 640 * 480. These standard templates are rotated and scaled, and some Gamma color noise is added to obtain adequate training instances. We use a data augmentation technique that combines the generated images and original images in the benchmark. The final network input is image patches with a fixed size of 640 * 480. We start training networks from scratch. The loss function of the pixel branch and type branch is Softmax loss, while the bbox branch uses L1 loss on the bounding box regression.

3.5 Testing

We adopt the same multi-scale testing and cropping strategy as used in the testing phase. The input testing images of the TT100K benchmark are 1280 * 1280, and the detection results are projected onto the original image according to the scale. In the projected images, we apply a bounding box merge method to remove redundant images.

4 Experiments

We describe the experimental details in this section. First, we provide a brief introduction to the two benchmarks. Then, the comparative experiments are introduced in detail.

4.1 Datasets and Evaluation Metrics

Tsinghua-Tencent 100K [16] is a large traffic sign benchmark that consists of 10,000 images collected from 10 regions of 5 cities in China (including both downtown regions and suburbs for each city). A total of 30,000 traffic sign instances are included in this benchmark, with detailed annotations for each sign, such as its bounding box, pixel mask and class. Each image is 2048 * 2048 pixels in size, which is larger than the high-resolution images considered in [7]. Common traffic conditions, such as illumination changes and weather changes, are included in this benchmark, but a data imbalance exists among classes.

Following [16], we ignore classes with fewer than 100 instances and implement our method on the remaining 45 classes. The evaluation metrics include accuracy and recall. Since many small traffic signs exist in the benchmark, the performance is also evaluated using the same detection metrics as used for the Microsoft COCO benchmark. The performance is evaluated on objects of different size, including small objects (area < area 96 * 96). The CTSD benchmark contains 1100 images (700 for training and 400 for testing) of different sizes (typical sizes are 1024 * 768 and 1280 * 720). The benchmark has 48 classes, and the proportion of traffic signs is relatively large. In this paper, we focus on the more challenging TT100K benchmark and utilize CTSD as a supplement to evaluate the generalizability of our model.

4.2 Experimental Details

Our experiments are divided into two parts to evaluate the performance of the proposed method. In the first part, we test the performances of several classic models on the TT100K dataset to identify the weaknesses of directly applying these models to TSD and TSR. The classic models used in our work include Fast-RCNN [33], Faster-RCNN [15], YOLO [34], and R-FCN [14]. The experimental results of Fast-RCNN and Faster-RCNN on TT100K are published in [16] and [40]. We focus only on YOLO and R-FCN in this section. In the second part, our experiments are based on the three design points mentioned above. The first is the feature extraction network, the second is the feature fusion strategy and the third is the hierarchical multi-task learning strategy. We initially convert the TT100K dataset into VOC format and fine-tune the YOLO model and R-FCN model. The results are shown in Table 3, which reveals that these two traditional models have poor performance in terms of small traffic sign detection.

YOLO [34] is a fast object-detection framework that does not require region proposal generation. We choose the model that is pre-trained on the Pascal VOC dataset and fine-tune it using the TT100K benchmark. The network is trained over 45,000 iterations. The images are resized to 416 * 416 and sent to the network. The network has a stride of 32 pixels; thus, small traffic signs cannot be effectively expressed in the final feature map, resulting in poor detection results for small objects. R-FCN [14] performs well on the Pascal VOC dataset. We choose ResNet-101 as the basic feature extraction network and initialize our network with a model that is pre-trained on ImageNet. Since ResNet-101 is deeper, we train the network over 200,000 iterations.

R-FCN performs poorly on TT100K. One reason is that TT100K is not as large as ImageNet; another reason is that small traffic signs cannot be detected easily. However, it outperforms YOLO on TT100K.

The above experimental results show that the general object-detection models do not perform well in terms of small sign detection. Taking into account the difficulties of TT100K, the general object-detection models are not applicable for this benchmark. In the second part, we propose several improvements to address these problems.

For the feature extraction network, our intuitive idea is to replace the feature extraction network with the VGG-16 net [27] and ResNet-50 [15] while retaining the three branches. We use the model pre-trained on ImageNet to fine-tune the two networks on TT100K. The VGG-16 network is trained over 80,000 iterations, and the ResNet-50 network is trained over 100,000 iterations. We propose a shallow network similar to VGG net with smaller (3 * 3) and fewer convolution filters in most layers to achieve a 75% decrease in the number of model parameters. This network is regarded as a baseline in our experiment. The detection results are shown in Table 4. From the table, we can see that our model significantly improves the accuracy compared to before.

To verify the effectiveness of the feature fusion strategy, we branch multi-task learning on feature maps of different sizes. The results are shown in Table 5. In the "Our-conv3" model, the multi-task learning network is connected to convolution conv3. The feature fusion strategy is added to the "Our-conv34" model. From the results, we can see that the feature fusion strategy can increase recall and accuracy at the same time. Based on the "Our-conv34" model, we further add the hierarchical multi-task learning strategy to our multi-task learning network. The results are shown in Table 6. The model "Ours" is the proposed model. Based on the strategy, we further increase the recall rate, especially the recall rate of small traffic signs.

Table 3. Detection results of different universal object detection models (%)

Model	YOLO	R-FCN
mAP	33	43

Table 4. Comparison of the simultaneous detection and classification results of different feature extraction networks on TT100K. (R): Recall, (A): Accuracy

Object size	(0, 32]	(32, 96]	(96, 400]	(0, 400]
Zhu et al. [1] (R)	0.87	0.94	0.88	0.91
Zhu et al. [1] (A)	0.82	0.91	0.91	0.88
VGG-16 [41] (R)	0.88	0.92	0.87	0.90
VGG-16 [41] (A)	0.73	0.88	0.87	0.82
ResNet-50 [18] (R)	0.59	0.65	0.57	0.62
ResNet-50 [18] (A)	0.55	0.61	0.59	0.58
Ours-baseline (R)	0.86	0.96	0.90	0.92
Ours-baseline (A)	0.86	0.94	0.94	0.91

Table 5. Comparison of the simultaneous detection and classification results of the feature fusion strategy on TT100K. (R): Recall, (A): Accuracy

Object size	(0, 32]	(32, 96]	(96, 400]	(0, 400]
Ours-baseline (R)	0.86	0.96	0.90	0.92
Ours-baseline (A)	0.86	0.94	0.94	0.91
Ours-conv3 (R)	0.89	0.96	0.93	0.93
Ours-conv3 (A)	0.85	0.94	0.93	0.90
Ours-conv34 (R)	0.88	0.97	0.92	0.93
Ours-conv34 (A)	0.86	0.95	0.94	0.91

Table 6. Comparison of the simultaneous detection and classification results of the hierarchical multi-task learning strategy on TT100K. (R): Recall, (A): Accuracy

Object size	(0, 32]	(32, 96]	(96, 400]	(0, 400]
Ours-baseline (R)	0.86	0.96	0.90	0.92
Ours-baseline (A)	0.86	0.94	0.94	0.91
Ours-conv34 (R)	0.88	0.97	0.92	0.93
Ours-conv34 (A)	0.86	0.95	0.94	0.91
Ours (R)	0.90	0.97	0.93	0.94
Ours (A)	0.86	0.94	0.95	0.91

Table 7. Comparison of the detection performance of the latest published literature on TT100K. (R): Recall, (A): Accuracy

Object size	(0, 32]	(32, 96]	(96, 400]	(0, 400]
Fast R-CNN [16] (R)	0.24	0.74	0.96	-
Fast R-CNN [16] (A)	0.45	0.51	0.55	-
Faster R-CNN [40] (R)	0.50	0.84	0.91	-

(*continued*)

Table 7. (*continued*)

Object size	(0, 32]	(32, 96]	(96, 400]	(0, 400]
Faster R-CNN [40] (A)	0.24	0.66	0.81	-
Zhu et al. [16] (R)	0.87	0.94	0.88	0.91
Zhu et al. [16] (A)	0.82	0.91	0.91	0.88
Zhu et al.-fast [16] (R)	0.74	0.91	0.86	0.74
Zhu et al.-fast [16] (A)	0.79	0.90	0.91	0.87
Perceptual GAN [40] (R)	0.89	0.96	0.89	0.93
Perceptual GAN [40] (A)	0.84	0.91	0.91	0.88
SOS-CNN [39] (R)	-	-	-	0.93
SOS-CNN [39] (A)	-	-	-	0.90
Ours-fast (R)	0.82	0.97	0.94	0.91
Ours-fast (A)	0.85	0.94	0.93	0.90
Ours (R)	**0.90**	**0.97**	**0.93**	**0.94**
Ours (A)	**0.86**	**0.94**	**0.95**	**0.91**

Table 8. Comparison of the detection performance on CTSD. (R): Recall, (A): Accuracy (%)

Model	Recall	Accuracy
Yang et al. [17]	-	**98.77**
Zhu et al. [16]	91.38	95.21
Ours	**95.40**	97.46

4.3 Performance Comparison

The newest detection results on TT100K are reported in Table 7. In the testing phase, the default testing scales are 1×, 2× and 4× the source image. The fast version is tested on only two scales, i.e., 1× and 2×, and the fast version is 4 times faster than the normal model. The results show that our proposed method achieves a recall rate of 94% and an accuracy of 91%, which are better than the results reported in [1–3]. Therefore, our proposed method achieves state-of-the-art results on TT100K. The precision-recall curves for different traffic sign sizes are shown in Fig. 7. Clearly, the performance of our approach on the three subsets has largely improved compared to before. The accuracy and the recall rate for each category for a Jaccard similarity coefficient of 0.5 are shown in Table 4, Table 5 and Table 3. We achieve nearly perfect results in several categories, such as “il100”, “pl100” and “pl120”. Some of the detection results for the TT100K testing set are shown in Fig. 9. Small traffic signs under different road conditions can be effectively detected by our proposed method, and our model can successfully detect the traffic signs that Zhu et al. [1] missed or detected incorrectly.

4.4 Generalization Experiments

We evaluate our method using the CTSD benchmark to verify the generalization. The images in the dataset have different sizes (typical sizes are 1024 * 768 and 1280 * 720); therefore, we resize the images to a fixed resolution (1024 * 1024) in the training and testing phases. We retrain the model in [1] and the proposed model on the CTSD. The model pre-trained on TT100K is used to initialize the network. We train both models over 50,000 iterations, and the results are shown in Table 8. From the results, we can see that our model also performs better than before on the CTSD. There are only 700 images available for training and 48 categories of traffic signs that need to be classified; thus, it is not suitable to train a CNN-based network on these data. Nevertheless, we still obtain results comparable with those of [11]. Our model achieves a recall rate of 95% and an accuracy of 97%. The precision-recall curves of the detection performance are shown in Fig. 8, and the detection results are shown in Fig. 9. In addition, we also collected some images in the field, the detection results for which are shown in Fig. 9. The results show the robustness of our model.

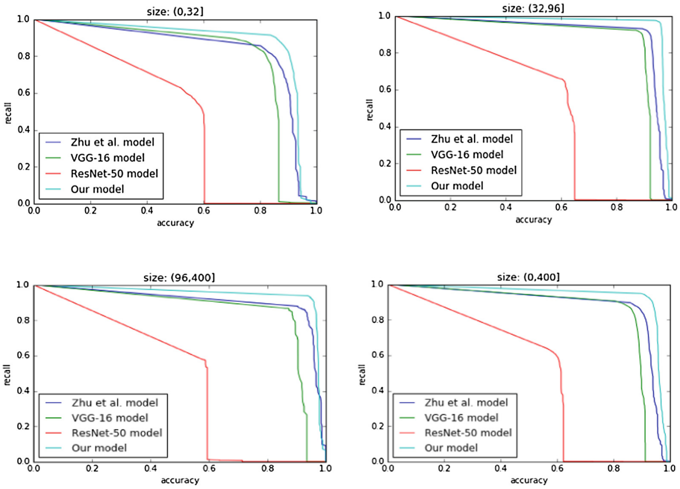


Fig. 7. Comparison of the overall detection performances on Tsinghua-Tencent 100K for small, medium and large traffic signs

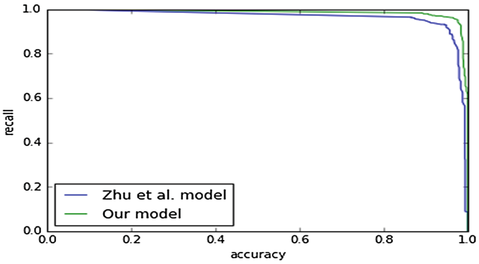


Fig. 8. The precision-recall curves of the detection performance on the CTSD

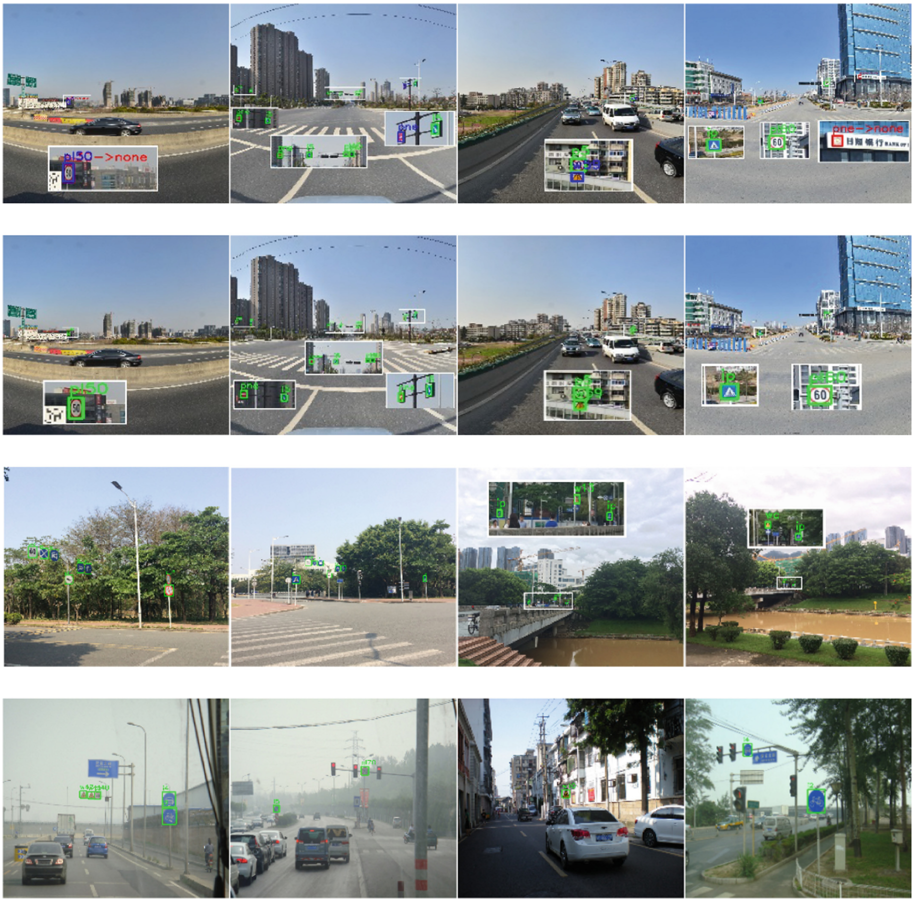


Fig. 9. The detection results on TT100K and the CTSD. The first row shows the detection results of Zhu et al. [1] on TT100K. The second row shows the detection results of our model on TT100K. The third row shows the detection results for our collected images. The fourth row shows the detection results of our model on the CTSD. The green, red and blue rectangles denote true positives, false positives and false negatives, respectively. (Color figure online)

4.5 Time Performance Analysis

The limitation of our model is its time performance. Because of the use of multi-scale and sliding window strategies during the testing phase, our model cannot meet the realtime requirements. For an original image with a resolution of 2048 * 2048, the processing speed can reach 0.25 fps. In this case, we can almost detect traffic signs at a distance of 100 m, which is important under the urban road environment. For the images of the CTSD, the resolution is smaller, and the processing speed can reach 1 fps.

5 Conclusions

In this paper, we aim to address the problem of small traffic sign detection and recognition in high-resolution images. To achieve this goal, we choose a multi-task learning model and make an improvement. First, we propose a feature extraction network with fewer parameters that is better than before. Second, we propose the feature fusion strategy to solve the problem of traffic sign detection for different sizes. Third, we propose a hierarchical multi-task learning strategy to improve the recall and accuracy. Our proposed model performs well in terms of small sign detection. To evaluate the performance of our model, we choose a challenging benchmark, TT100K, which contains high-resolution images with many small traffic signs. It is difficult for CNNs to process large images with a resolution of 2048 * 2048. To address this problem, the entire image is cropped into patches of fixed size. Our method generates relatively large feature maps that contain more contextual information and a larger receptive field. A multi-scale training and testing strategy is also applied. To verify the generalization of the model, we evaluate it on another benchmark.

The shortcoming of our proposed method is that it cannot be used for real-time applications. In the future, we will make the method more efficient.

Acknowledgment. This research was supported in part by the Shenzhen Science and Technology Innovation Commission research project JCYJ20170307151831260.

Ke Yu is the co-first author and contributed equally with You Lei. Wang Hongpeng is the corresponding author, his email is wanghp@hit.edu.cn.

References

1. Gudigar, A., Chokkadi, S., Raghavendra, U.: A review on automatic detection and recognition of traffic sign. Multimedia Tools Appl. **75**(1), 333 (2016)
2. Sermanet, P., LeCun, Y.: Traffic sign recognition with multi-scale convolutional networks. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 2809–2813. IEEE (2011)
3. Ciresan, D., Meier, U., Masci, J., et al.: A committee of neural networks for traffic sign classification. In: The 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1918–1921. IEEE (2011)
4. Ciresan, D., Meier, U., Masci, J., et al.: Multi-column deep neural network for traffic sign classification. Neural Netw. **32**, 333–338 (2012)
5. Jin, J., Fu, K., Zhang, C.: Traffic sign recognition with hinge loss trained convolutional neural network. IEEE Trans. Intell. Transp. Syst. **15**(5), 1991–2000 (2014)
6. Houben, S., Stallkamp, J., Salmen, J., et al.: Detection of traffic signs in realworld images: the German Traffic Sign Detection Benchmark. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2013)
7. Stallkamp, J., Schlipsing, M., Salmen, J., et al.: Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw. **32**, 323–332 (2012)
8. Ellahyani, A., El Ansari, M., El Jaafari, I.: Traffic sign detection and recognition based on random forests. Appl. Soft Comput. **46**, 805–815 (2016)

9. Lillo-Castellano, J.M., Mora-Jimnez, I., Figuera-Pozuelo, C., et al.: Traffic sign segmentation and classification using statistical learning methods. Neurocomputing **153**, 286–299 (2015)
10. Mogelmose, A., Trivedi, M.M., Moeslund, T.B.: Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. IEEE Trans. Intell. Transp. Syst. **13**(4), 1484–1497 (2012)
11. Barnes, N., Zelinsky, A., Fletcher, L.S.: Real-time speed sign detection using the radial symmetry detector. IEEE Trans. Intell. Transp. Syst. **9**(2), 322–332 (2008)
12. Loy, G., Barnes, N.: Fast shape-based road sign detection for a driver assistance system. In: Proceedings 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), vol. 1, pp. 70–75. IEEE (2004)
13. Liu, Wei, et al.: SSD: single shot multibox detector. In: Leibe, Bastian, Matas, Jiri, Sebe, Nicu, Welling, Max (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
14. Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems, pp. 379–387 (2016)
15. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
16. Zhu, Z., Liang, D., Zhang, S., et al.: Traffic-sign detection and classification in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2110–2118 (2016)
17. Yang, Y., Luo, H., Xu, H., et al.: Towards real-time traffic sign detection and classification. IEEE Trans. Intell. Transp. Syst. **17**(7), 2022–2031 (2016)
18. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
19. Han, Y., Virupakshappa, K., Oruklu, E.: Robust traffic sign recognition with feature extraction and k-NN classification methods. In: 2015 IEEE International Conference on Electro/Information Technology (EIT), pp. 484–488. IEEE (2015)
20. Salti, S., Petrelli, A., Tombari, F., et al.: Traffic sign detection via interest region extraction. Pattern Recogn. **48**(4), 1039–1049 (2015)
21. Ruta, A., Li, Y., Liu, X.: Real-time traffic sign recognition from video by class-specific discriminative features. Pattern Recogn. **43**(1), 416–430 (2010)
22. Ellahyani, A., El Ansari, M.: Mean shift and log-polar transform for road sign detection. Multimed. Tools Appl. **76**, 1–19 (2016)
23. Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., et al.: Roadsign detection and recognition based on support vector machines. IEEE Trans. Intell. Transp. Syst. **8**(2), 264–278 (2007)
24. Chen, T., Lu, S.: Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression. IEEE Trans. Veh. Technol. **65**(6), 4006–4015 (2016)
25. Wang, G., Ren, G., Wu, Z., et al.: A robust, coarse-to-fine traffic sign detection method. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–5. IEEE (2013)
26. Zaklouta, F., Stanciulescu, B.: Real-time traffic sign recognition in three stages. Rob. Auton. Syst. **62**(1), 16–24 (2014)
27. Greenhalgh, J., Mirmehdi, M.: Real-time detection and recognition of road traffic signs. IEEE Trans. Intell. Transp. Syst. **13**(4), 1498–1506 (2012)
28. Berkaya, S.K., Gunduz, H., Ozsen, O., et al.: On circular traffic sign detection and recognition. Expert Syst. Appl. **48**, 67–75 (2016)

29. Wang, G., Ren, G., Wu, Z., et al.: A hierarchical method for traffic sign classification with support vector machines. In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2013)
30. Huang, Z., Yu, Y., Gu, J., et al.: An efficient method for traffic sign recognition based on extreme learning machine. IEEE Trans. Cybern. **47**(4), 920–933 (2017)
31. Liu, C., Chang, F., Chen, Z., et al.: Fast traffic sign recognition via high contrast region extraction and extended sparse representation. IEEE Trans. Intell. Transp. Syst. **17**(1), 79–92 (2016)
32. Mathias, M., Timofte, R., Benenson, R., et al.: Traffic sign recognition - how far are we from the solution? In: The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2013)
33. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
34. Redmon, J., Divvala, S., Girshick, R., et al.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
35. Huval, B., Wang, T., Tandon, S., et al.: An empirical evaluation of deep learning on highway driving. arXiv preprint. arXiv:1504.01716 (2015)
36. Sermanet, P., Eigen, D., Zhang, X., et al.: OverFeat: integrated recognition, localization and detection using convolutional networks. Eprint arXiv (2013)
37. Zhu, Y., Zhang, C., Zhou, D.: Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing **214**, 758–766 (2016)
38. Hu, P., Ramanan, D.: Finding tiny faces. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1522–1530. IEEE (2017)
39. Meng, Z., Fan, X., Chen, X., et al.: Detecting small signs from large images. arXiv preprint. arXiv:1706.08574 (2017)
40. Li, J., Liang, X., Wei, Y., et al.: Perceptual generative adversarial networks for small object detection. In: IEEE CVPR (2017)
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for largescale image recognition. arXiv preprint. arXiv:1409.1556 (2014)
42. Liu, C., Chang, F., Chen, Z.: Rapid multiclass traffic sign detection in high resolution images. IEEE Trans. Intell. Transp. Syst. **15**(6), 2394–2403 (2014)