Real-Time Traffic Sign Recognition Based on Efficient CNNs in the Wild

Jia Li and Zengfu Wang, *Member, IEEE*

***Abstract*— Both unmanned vehicles and driver assistance systems require solving the problem of traffic sign recognition. A lot of work has been done in this area, but no approach has been presented to perform the task with high accuracy and high speed under various conditions until now. In this paper, we have designed and implemented a detector by adopting the framework of faster R-convolutional neural networks (CNN) and the structure of MobileNet. Here, color and shape information have been used to refine the localizations of small traffic signs, which are not easy to regress precisely. Finally, an efficient CNN with asymmetric kernels is used to be the classifier of traffic signs. Both the detector and the classifier have been trained on challenging public benchmarks. The results show that the proposed detector can detect all categories of traffic signs. The detector and the classifier proposed here are proved to be superior to the state-of-the-art method. Our code and results are available online.**

***Index Terms*— Traffic sign recognition, Faster R-CNN, localization refinement, efficient CNN.**

I. Introduction

THE problem of real-time traffic sign recognition in the wild has to be solved to meet the growing need of automatic driving. The traffic sign recognition usually consists of two steps: traffic sign detection and traffic sign classification. A lot of work has been done to complete the task [1]–[20]. Most of them have trained their models on the German Traffic Sign Detection Benchmark (GTSDB) database [1] (see Figure 1) and the German Traffic Sign Recognition Benchmark (GTSRB) database [2]. Accuracy and speed are surely the two main requirements in practical applications. Although lots of relevant approaches have been presented in previous work, no one can solve the traffic sign recognition problem very well in conditions of different illumination, motion blur, occlusion and so on. So, more effective and more robust approaches need to be developed.

As for the task of traffic sign detection, existing approaches have achieved good results on some benchmark databases. Next to 100% recall is claimed in some existing work and it seems that the problem has been solved. Unfortunately, the problem has not been solved and has remained difficult for the limited representation of databases. The GTSDB database, which is mostly used, only collects 600 training images and 300 evaluation images, among which merely three categories of traffic signs are evaluated.

Fig. 1. Examples of traffic scenes in the GTSDB database.

In most of previous work, traditional machine learning based approaches, such as probability model, HOG features and support vector machine [3] are just designed to detect specific categories of traffic signs. Considering the inherent color and shape information of traffic signs, Berkaya *et al.* [4] use a circle detection algorithm and an RGB-based color threshold technique to detect circular traffic signs. However, neither intuitive color features nor intuitive shape features are reliable in the wild. We hold the opinion that the detection task can be solved using convolutional neural networks (CNN) if there are abundant training images, which is mainly discussed in this paper.

As for the task of traffic sign classification, convolutional neural networks achieve the state-of-the-art [5]–[7], [20], thanks to the abundant GTSRB database. It should be noted that most of existing work performs the detection task and the classification task separately, which is unrealistic in applications, for there is a gap between the two tasks. Previous work about traffic sign detection, especially that based on machine learning approaches (CNN, for instance), does not always give a compact bounding box, making the subsequent classification task more difficult [21]. The localization refinement may need to be done before classification, which is considered in this study.

Our contributions in this paper are summarized as follows:

- Faster R-CNN [22] and MobileNets [23] are combined and modified to make the detection process more efficient. A pretty good detection result of unique traffic signs is reported for the first time on the GTSDB database.

Manuscript received November 27, 2017; revised April 15, 2018; accepted May 7, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61472393. The Associate Editor for this paper was S. S. Nedevschi. *(Corresponding author: Zengfu Wang.)*

J. Li is with the Department of Automation, University of Science and Technology of China, Hefei 230026, China (e-mail: jialee@mail.ustc.edu.cn).

Z. Wang is with the Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China, and also with the National Engineering Laboratory for Speech and Language Information Processing, Hefei 230026, China (e-mail: zfwang@ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2018.2843815

- An easy approach based on color and shape information for the localization refinement of small traffic signs is proposed, improving the detection quality and the classification accuracy of typical traffic signs.
- An efficient and robust CNN with asymmetric kernels is designed to be the classifier. Besides, self-comparison is done to demonstrate the strength of the designed structure.
- The models designed here can be trained end-to-end and the training process is described in detail. Some tricks are used to overcome the non-convergence and overfitting problems of CNN when only a small amount of data is available.
- The proposed approach is up to real-time applications. The source code as well as the trained CNN models are released online[1] for future researchers.

II. Related Work

A. Traffic Sign Detection

The task of traffic sign detection is very challenging due to quite a few disturbances, such as nonuniform illumination, motion blur, occlusion and hard negative samples. Manual features, such as HOG, enhancement information of typical color or geometric shape, tend to fail in many difficult environments. However, CNNs are considered to be robust and powerful in many applications and some relevant work based on CNNs has been done. Zhu *et al.* [8] propose a novel framework with two deep learning components including fully convolutional network (FCN) guided traffic sign proposals and deep convolutional neural network (CNN) for object classification. The proposed approach in [8] is experimentally compared with R-CNN [24]. Zhu *et al.* [9] propose a multi-class fully convolutional network to simultaneously detect and classify traffic signs. In addition, they propose another detection network treating all traffic signs as one category in the detection step. The performance of original Fast R-CNN [25] for traffic sign recognition is analyzed in [9]. Peng *et al.* [10] use original Faster R-CNN [22] to perform the task and the results show that this approach is promising, but the accuracy and speed are very disappointing.

Most recently, popular detection frameworks, such as Faster R-CNN, YOLO [26] and SSD [27], are widely studied for generic detection task, among which Faster R-CNN is relatively better at detecting small objects. The traffic sign usually occupies only a small portion of the entire traffic scene image. Sizes of traffic signs vary a large range, usually $16 \times 16 \sim 128 \times 128$ (pixels) in the GTSDB database, which is determined by the distance from traffic signs to the camera. Thus, the framework of Faster R-CNN is adopted. MobileNets [23] which use depthwise separable convolutions, are efficient models for mobile and embedded vision applications. We use the structure of MobileNets to build the bottom layers of our detector and adopt the depthwise separable convolutions in the classification head of Faster R-CNN, making the detection process faster.

[1] https://github.com/USTClj/Traffic_Sign_Recognition_Efficient_CNNs

B. Localization Refinement

A well designed and trained CNN is able to classify traffic signs whose bounding boxes are regressed approximately. However, the classification accuracy declines obviously when the bounding box fails to enclose the traffic sign precisely. A refinement step can significantly improve the detection quality, leading to a higher accuracy [21]. Traditional computer vision techniques such as color threshold and shape detection can be used to locate the traffic sign very precisely (or perhaps a fail detection occurs) from the whole image captured by an onboard camera [4], [28]–[30]. But these approaches work for circular signs with typical colors only and cannot be generalized to other shapes and colors. It should be noted that these approaches are easy to be influenced heavily by environmental conditions such as illumination and hard negative samples with traffic-sign-like appearances such as tail lights and advertising signs. In our study, we utilize the shape and color priors of traffic signs to refine the localizations, making the detection results more precise.

C. Traffic Sign Classification

Benefitting from the abundant GSTRB database, CNNs achieve the highest classification accuracy [2]. Multi-Column Deep Neural Network (MCDNN) [5] achieves the accuracy of 99.4% on the evaluation images. Multi-scale CNN [6] achieves the accuracy of 99.17%. Hinge loss trained CNN [7] achieves nearly perfect accuracy of 99.65%. These approaches have achieved very high accuracy, but they are time-consuming or need to learn large amounts of parameters. The CNN with learned color and spatial transformation [31] achieves a satisfying accuracy of 99.59%, and the prediction is pretty fast, but the designed structure is complex. In this study, a simple but very powerful CNN is presented, which is up to real-time applications. The proposed network is fast to implement and train. The purpose of designing such a classifier here is mainly aimed at building a complete pipeline of traffic sign recognition.

III. Approach

In this section, we present the details of our traffic sign recognition pipeline, i.e. the traffic sign detection step, the localization refinement step and the traffic sign classification step, which is illustrated in Figure 2. In this study, the detector treats all traffic signs as one category in the detection step. The classifier predicts the labels of candidate traffic signs. The localization refinement algorithm is designed and performed to further improve the precision of localization and the accuracy of classification.

A. Traffic Sign Detection

We adopt the framework of Faster R-CNN and the MobileNet as the backbone CNN for traffic sign detection. MobileNets, which use depthwise separable convolutions, are efficient models. The depthwise separable convolution factorizes a standard convolution into a depthwise convolution and a 1×1 pointwise convolution [23]. In this study, the structure of

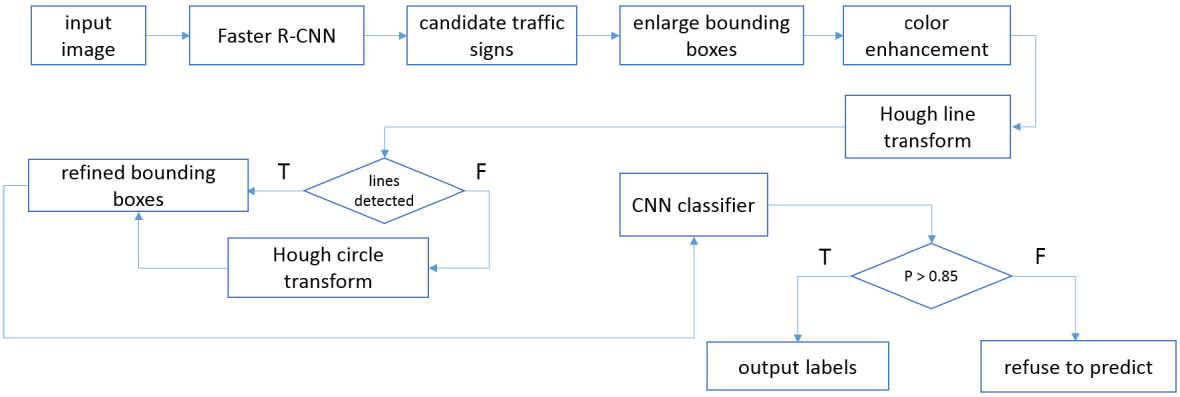

Fig. 2. The designed pipeline of traffic sign recognition. The block named Faster R-CNN represents the detection step, and the block named CNN classifier represents the classification step. The rest of blocks represent the procedures of localization refinement step. The CNN classifier will refuse to output the label of the candidate traffic sign if its confidence probability is less than 0.85.

MobileNet is adopted to build the base convolutional layers in Faster R-CNN. Besides, we absorb the structure of depthwise separable convolution and use this kind of convolution to build the classification head of Faster R-CNN. The backbone CNN (the base convolutional layers) takes as input the whole traffic scene image and produces the feature map of the input image.

The region proposal network (RPN) and the classifier share the base convolutional layers. The RPN generates proposals from the feature map outputted by the base convolutional layers and some ROIs (region of interest) are obtained. The RoI pooling layer converts features inside any valid region of interest into a fixed size feature map. After that, the fixed size feature map is fed into the classifier to output the detection result.

The RPN and the classifier have similar functions but they play different roles. The RPN is able to distinguish between the foreground and the background and regress the bounding boxes of proposals, while the classifier is able to predict the confidence probabilities of traffic signs and regress the refined bounding boxes of candidate traffic signs. We will discuss the components of our detector in detail next.

1) The Base Convolutional Layers: Supposing the size of input image is 800 × 1360, the structure of the base convolutional layers is defined in Table I.

Where

"Conv"	denotes a standard convolution,
"Conv dw"	denotes a depthwise convolution,
"s2"	denotes that the convolution stride is 2 × 2,
"s1"	denotes that the convolution stride is 1 × 1.

For each depthwise separable convolution operation, the depthwise convolution is used first to apply a single filter per each input channel (input depth). The pointwise convolution is then used to combine the output of the depthwise layer linearly in order to generate the same number of features as a standard convolution dose. The reduction of computational cost is in proportion to the number of output feature map channel and the square of kernel size. More details can be found in [23].

TABLE I
THE ARCHITECTURE OF BASE CONVOLUTIONAL LAYERS

Type / Stride	Kernel Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	800 × 1360 × 3
Conv dw / s1	3 × 3 × 32 dw	400 × 680 × 32
Conv / s1	1 × 1 × 32 × 64	400 × 680 × 32
Conv dw / s2	3 × 3 × 64 dw	400 × 680 × 64
Conv / s1	1 × 1 × 64 × 128	200 × 340 × 64
Conv dw / s1	3 × 3 × 128 dw	200 × 340 × 128
Conv / s1	1 × 1 × 128 × 128	200 × 340 × 128
Conv dw / s2	3 × 3 × 128 dw	200 × 340 × 128
Conv / s1	1 × 1 × 128 × 256	100 × 170 × 128
Conv dw / s1	3 × 3 × 256 dw	100 × 170 × 256
Conv / s1	1 × 1 × 256 × 256	100 × 170 × 256
Conv dw / s2	3 × 3 × 256 dw	50 × 85 × 256
Conv / s1	1 × 1 × 256 × 512	50 × 85 × 256
Conv dw / s1 5 × Conv / s1	3 × 3 × 512 dw 1 × 1 × 512 × 512	50 × 85 × 512

All layers in Table I are followed by a batch normalization layer and a ReLU layer. According to Table I, the size of the output feature map is 1/16 of the input size. It should be noted that the size of the input image does not have to be 800 × 1360 because the network is fully convolutional.

2) The Region Proposal Network (RPN): The region proposal network takes as input a 3 × 3 spatial window of the input feature map. An anchor is centered at each sliding-window location, which is illustrated in Figure 3. Three scales and three aspect ratios of anchors [22] are used in this study, which will be explained in Section IV-C.1. The anchor scales should be tuned according to the scales of traffic signs to detect because traffic signs usually occupy only a small fraction of the whole image and the sizes vary a large range.

The features in each sliding window are fed into a 3 × 3 convolutional layer and is mapped into a lower-dimensional feature (256-d). The number of sliding window locations is 50 × 85 according to Table I and 50 × 85 features which are 256-d are outputted by this layer.

Then, the 256-d features are fed into a two-layer fully convolutional neural network to yield 9 anchors. The two convolutional layers with sigmoid activation functions and

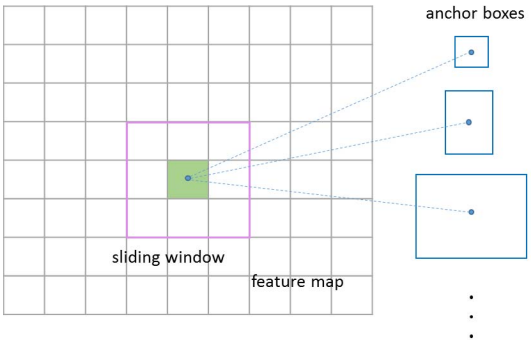

Fig. 3. Anchors. The region proposal network proposes several anchor boxes at each sliding position on the feature map. In this work, the number of anchor boxes is set to 9 at each sliding position for the trade-off between recall and processing speed.

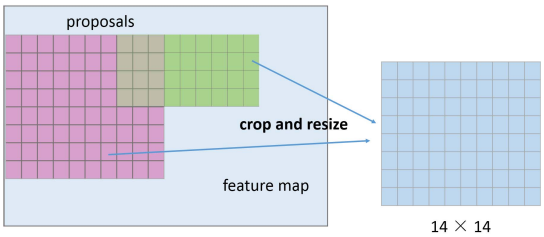

Fig. 4. The operation of RoI pooling. The feature map is outputted by the base convolutional layers. The fixed spatial extent is set to 14 × 14 according to the detection performances.

linear activation functions respectively serve as an anchor box score layer and an anchor box regression layer [22]. As a result, 9 possible proposals are yielded at each sliding position.

3) The RoI Pooling Layer: The RoI pooling layer [25] uses max pooling to convert the features inside any valid region of interest proposed by the RPN into a feature map with a fixed spatial extent. But the max pooling operation brings about the problem of misalignment obviously. A differentiable RoI warping layer [32] is added before the standard max pooling to perform the RoI pooling, compensating for this problem to some extent. However, for simplicity in our implementation, we perform the RoI pooling by cropping a feature region proposed by the RPN and resize the region to the fixed spatial extent of 14 × 14 via bilinear interpolation, which is illustrated in Figure 4. Subsequently, the fixed size feature map is fed into the classifier after the RPN. The classifier which will be introduced next is a small classification network to discriminate between traffic signs and negative samples.

4) The Classifier: The classifier after the RPN is defined in Table II. All convolutional layers are followed by a batch normalization layer and a ReLU layer. In Table II,"n_roi" denotes the number of ROIs to be selected and used, which are proposed by the RPN. The classifier has two fully connected layers (dense layer), i.e., a box classification layer and a box regression layer [25]. The first dense layer has two outputs, which are fed into the softmax layer to compute the confidence probabilities of being traffic signs and negative samples. The second dense layer with linear activation

TABLE II
THE CLASSIFIER ARCHITECTURE FOR TRAFFIC SIGN REGIONS

Type / Stride	Kernel Shape	Input Size
Conv dw / s2	3 × 3 × 512 dw	n_roi × 14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	n_roi × 7 × 7 × 512
Conv dw / s1	3 × 3 × 1024 dw	n_roi × 7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	n_roi × 7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	n_roi × 7× 7 × 1024
Dense	1024 × 2	n_roi × 1× 1 × 1024
Softmax	\	n_roi × 2
Dense	1024 × 4	n_roi × 1× 1 × 1024
Linear	\	n_roi × 4

functions regresses the bounding boxes of candidate traffic signs.

5) Loss Functions: The loss function and the parameterization of coordinates for bounding box regression in this work are the same as those in Faster R-CNN [22]. We set the balancing parameter λ to 1 in the loss function. The loss function of the RPN and the loss function of the classifier after the RPN share the same form but are optimized separately, which will be discussed in Section IV-C.1.

B. Localization Refinement

Generally speaking, it is easy to locate traffic signs which are larger than 40 × 40 precisely (see Figure 5). A positive detection result is only valid if the predicted bounding box has at least 0.5 overlap (IoU) with the ground truth. But small traffic signs cannot be localized precisely enough, making the subsequent classification task more difficult.

Zhu *et al.* [21] refine localizations via the energy function consists of color term, shape term and smoothness term, achieving very precise results. However, a simple nonparametric approach is proposed here to perform the task, meeting the need of most circumstances.

After the detection step, we compute the center of the regressed bounding box and enlarge its boundary to attempt to enclose the entire traffic sign region. Then, we refine the localization by the method of segmentation. Traffic signs usually have typical color information, i.e., red, blue and yellow and typical geometric information, i.e., circle, rectangle, triangle and regular octagon. The prior knowledge of color and shape can be used to refine the localizations. The normalized RGB color space (RGBN) is used to reduce the effects of linear illumination variations, allowing finding the correct thresholds [11]. The relationship between the RGB and RGBN color spaces is formulated as follows:

$$C' = \frac{C}{R+G+B}, C \in \{R, G, B\}. \tag{1}$$

Subsequently, red, blue and yellow color information are enhanced by a set of transformations [12]:

$$\begin{aligned} E(R) &= \max(0, \min(R-B, R-G)) \\ E(B) &= \max(0, \min(B-R, B-G)) \\ E(Y) &= \max(0, \min(R-B, G-B)). \end{aligned} \tag{2}$$

Fig. 5. Examples of detection results. Traffic signs which are larger than 40 × 40 are easy to regress precisely, while small traffic signs are difficult to regress precisely.

Fig. 6. The procedures of the localization refinement for red circular traffic signs. The original RGB patch is transformed into the RGBN color space and then the R, B, Y color information are enhanced. After that, the binary image is obtained via segmentation. The contour of the traffic sign is then extracted via morphological operations. Subsequently, the center and radius are calculated using Hough circle transform. Finally the localization of the traffic sign is refined.

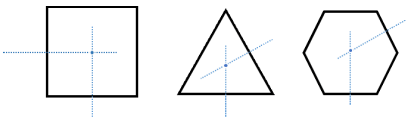

Fig. 7. Examples of detecting contours and centers of traffic signs.

We segment the traffic sign with specific color from the enlarged bounding box region using the enhanced color information. Taking a red traffic sign as an example, the segmentation threshold of red color is determined by Otsu's method [33] automatically. And a binary image is obtained after segmentation. Then, operations based on mathematical morphology are performed on the binary image and the contour of the traffic sign is extracted. For circular traffic signs, the center and radius are calculated via the method of Hough circle transform [34]. The procedure is illustrated in Figure 6.

For traffic signs which are not circular, similar procedures could be done, which is illustrated in Figure 7. Two longest lines, which are unparallel to each other, are detected from the binary image by Hough line transform [34]. After that, the centroid of geometric shape is obtained, which is used to refine the localization.

It should be noted that merely the localizations of traffic signs with red, yellow or blue color information can be refined using our approach, which is mostly encountered in practice. Since the proposed refinement algorithm is nonparametric, we detect contours on the enhanced R, B, and Y channel in turn. If two lines unparallel to each other are detected by Hough line transform, the Hough circle transform will not be executed, and vice versa.

The last but not least, the refined bounding box will be rejected if its IoU with the bounding box regressed by the detector is less than 0.5 to avoid a false refinement. The localization refinement step is helpful to further improve the accuracy of the subsequent classification task.

C. Traffic Sign Classification

Convolutional neural networks have achieved the state-of-the-art in traffic sign classification [13]. Though many approaches have been proposed, a fast and accurate CNN is presented here. The architecture designed in this study is very easy to implement and train. A 1 × n convolution and an n × 1 convolution are used to replace an n × n convolution [35], [36], decreasing both the number of kernel parameters and convolutional operations. For example, using a 3 × 1 convolution followed by a 1 × 3 convolution is equivalent to sliding a two layer network with the same receptive field as in a 3 × 3 convolution [36], saving 33% of the computational cost (see Figure 8). The computational cost saving increases dramatically as n grows. If we replace a 7 × 7 convolution with a 1 × 7 convolution and a 7 × 1 convolution, the number of kernel parameters and convolutional operations will decrease by (7 × 7 - 1 × 7 - 7 × 1) / (7 × 7) ≈ 71.4% (the spatial width and height of the feature map remain the same after convolution operation in our case). Besides, the classification accuracy improves in our experiments when we use different asymmetric convolution kernels to extract features.

The classifier is designed in Table III. All layers are followed by a batch normalization layer and a ReLU layer except for the final dense layer. The sixth layer extracts features on the same feature map outputted by the fifth layer with different kernels, learning different spatial information. Feature maps outputted by the inception module are concatenated along the channel axis to fuse different spatial information. The number of output in the last layer is 43 because we need to recognize 43 classes of traffic signs in the GTSRB database.

IV. Experiments

We implement the proposed CNNs using Keras with TensorFlow backend. We have trained and evaluated the proposed detector on the German Traffic Sign Detection Benchmark (GTSDB) database [1] and trained and evaluated the proposed classifier on the German Traffic Sign Recognition Benchmark (GTSRB) database [2]. We did the experiments on a computer with an NVIDIA GTX 1080 GPU. Two GB GPU memory is enough to train and run our models.

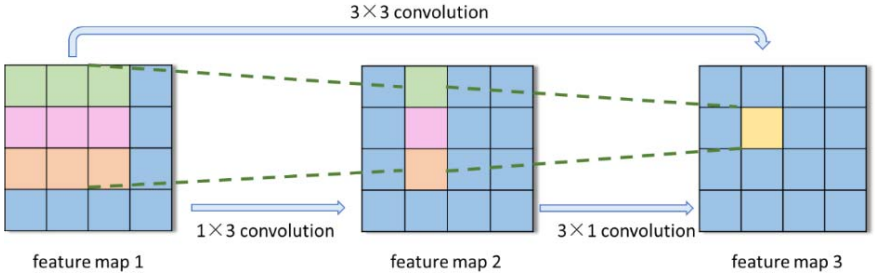

Fig. 8. Replacing a 3 × 3 convolution with a 1 × 3 convolution and a 3 × 1 convolution. Each neuron output in feature map 3 is under the influence of a 3 × 3 region in feature map 1.

TABLE III
THE CLASSIFIER ARCHITECTURE

Layer	Type / Stride	Kernel Shape	Output Size
1	Conv / s1	3 × 3	48 × 48 × 3
2	Conv / s1	7 × 1	48 × 48 × 3
3	Conv / s1	1×7	48 × 48 × 3
4	Max pool / s2	2 × 2	24 × 24 × 48
5	Dropout	\	24 × 24 × 48
Inception Module			
6-1	Conv / s1	3 × 1	24 × 24 × 64
7-1	Conv / s1	1 × 3	24 × 24 × 64
6-2	Conv / s1	1 × 7	24 × 24 × 64
7-2	Conv / s1	7 × 1	24 × 24 × 64
8	Concatenate	\	24 × 24 × 128
Concatenated Feature Maps			
9	Max Pool / s2	2 × 2	12 × 12 × 128
10	Dropout	\	12 × 12 × 128
11	Conv / s1	3 × 3	12 × 12 × 128
12	Conv / s1	3 × 3	12 × 12 × 256
13	Max Pool / s2	2 × 2	6 × 6 × 256
14	Dropout	\	6 × 6 × 256
15	Dense	\	256
16	Dropout	\	256
17	Dense - Softmax	\	43

A. Benchmarks and Data Preprocessing

The GTSDB database which is adopted in our work has been widely used [1], [3], [14]–[19] to train and evaluate traffic sign detectors. A much bigger traffic sign benchmark database called Tsinghua-Tencent 100K [9] was created in 2016 and was claimed to be superior to the GTSDB database. However, we hold the opinion that the GTSDB database is more challenging and more representative. Traffic scene images in the GTSDB database are extracted from video sequences, while original images of high quality in the Tsinghua-Tencent 100K database are captured by panoramic cameras at intervals of about 10 m for the purpose of creating street views. The majority of traffic scenes in the Tsinghua-Tencent 100K database are cleaner and contain fewer negative samples, such as vehicles, taillights, pedestrians and traffic lights. Besides, the database does not take account of the disturbance of motion blur, fogy or rainy weather.

The sizes of traffic signs to detect in the GTSDB database vary from 16 × 16 to 128 × 128 and the size of traffic scenes is 800 × 1360. There are 600 training images and 300 evaluation images in this database. We do data augmentation by horizontal flip, getting 1200 training images. The traffic signs in the GTSDB database can be divided into four categories, i.e., prohibitory signs with red color and circular shape, danger signs with red color and triangular shape, mandatory signs with blue color and circular shape, and the rest of traffic signs called unique traffic signs with different shapes and colors (see Figure 9). In the evaluation set, there are 92 unique traffic signs (there are 3 additional traffic signs which were not annotated in the original database) to detect.

Fig. 9. Examples of unique traffic signs [37].

Fig. 10. Examples of cropped and resized traffic signs. The indexes above the images are the labels of traffic signs.

The classifier is trained and evaluated on the GTSRB database with 39209 training images and 12630 evaluation images. There are 43 classes of traffic signs in this database and the sizes of images vary from 15 × 15 to 250 × 250. In the process of data preprocessing, all traffic signs are cropped and resized to the fixed spatial extent of 48 × 48, which is shown in Figure 10. One image is chosen randomly among every 30 frames in the training set. As a result, a validation subset is created, containing 1307 traffic sign images.

B. Evaluation Metrics

A positive proposal is considered to be true if its Jaccard similarity (IoU between the predicted bounding box and the annotated bounding box) is greater than 0.5. Recall and precision are used as the metrics to evaluate the detection performance. Accuracy is used as the metric to evaluate the classification performance.

C. Training

1) Training the Detector: Training images are fed into the detection network one by one. For each image, 256 anchors

mapped from the ground-truth bounding boxes are randomly sampled, among which the ration of negative anchors and positive anchors is set to 1 : 1. Generally speaking, there are more negative anchors than positive anchors and positive anchors are duplicated for the balance. The anchor boxes which cross image boundaries are ignored, so that they do not contribute to the loss [22]. Three anchor box scales of 24^2, 34^2 and 54^2 and three anchor box ratios of 1 : 1, 1 : 2 and 2 : 1 are used for the trade-off between recall and processing speed. In fact, we have used different sets of scales and ratios to train the same network during training process, boosting the performance of detection. To simplify the training process, alternating training [22] is adopted.

We fine-tune the base convolutional layers, a MobileNet, on the GTSDB database. Here, initial parameters of MobileNets, which are available at [38], have been pre-trained on the ImageNet database [39]. In addition, the weights (means and standard deviations) of each batch normalization layer in the backbone CNN have been computed on the ImageNet database, and they are frozen during the training on the GTSDB database. This trick makes our detector easier to train.

The region proposal network (RPN) and the classifier share the base convolutional layers and are trained by turns. When the RPN is trained on a mini-batch, the parameters of the RPN and the base convolutional layers are updated once. After that, the RPN generates many proposals, among which 256 proposals are selected via non-maximum suppression (NMS). The proposal whose region overlaps a ground truth region more than 70% is regarded as a positive proposal. Otherwise, it is regarded as a negative proposal. Subsequently, we use these selected positive and negative samples to train the classifier of the detector. The parameters of the classifier are updated once and the parameters of the base convolutional layers are updated once again. This process is iterated and the entire detection network is optimized. The Adam algorithm [40] is used for the optimization of the loss functions. The initial learning rates of the RPN and the classifier are all set to 1e-4 with the learning rate decay of 4e-5 per mini-batch. We trained the network for 300 epochs totally.

2) Training the Classifier: We train the classifier on the GTSRB database. The dropout rates in Table III are set to 0.2, 0.2, 0.3 and 0.4 respectively. The Adam algorithm with the initial learning rate of 0.001 is used. The learning rate decay is set to 1e-6 per mini-batch, and the batch size is set to 16. We trained the classifier for 30 epochs first without data augmentation. After that, we trained the classifier for 200 additional epochs with data augmentation such as random rotation ([-10, 10] degrees), scaling ([0.8, 1.2] ration), horizontal and vertical shift ([-4, 4] pixels), and random shear transformation ([0, 0.1] radians).

V. Evaluation

A. The Performance of Detection

When we apply the well-trained detector, some hyperparameters should be modified for the trade-off between recall and prediction speed. Only 64 proposals predicted by the RPN are selected via non-maximum suppression with the overlap threshold of 0.7, and then these proposals are fed into the classifier. The classifier regards a candidate object, whose confidence probability is bigger than 0.8 (a threshold set manually), as a traffic sign and outputs its bounding box. It should be noted that the detection result does not depend much on the confidence probability threshold, for the reason that the confidence probability of a negative example tends to be low according to the results. Some detection results are shown and analyzed in Figure 11. As mentioned above, small traffic signs are not easy to regress precisely, which is shown in Figure 11. The localization refinement algorithm designed in Section III-B can be used to future improve the precision of coordinate regression, however it is designed for traffic signs with specific shapes and colors.

Only three categories of traffic sign (excluding the unique category which is illustrated in Figure 9) are evaluated in previous work [3], [4], [14], [16]–[18]. Approaches proposed by [8], [9], and [14] were claimed to be able to detect unique traffic signs, while approaches proposed by [3], [4], [13], and [15] failed. Our detector can detect unique traffic signs, and the performance of detecting unique traffic signs has been evaluated as well. Table IV shows a comparison to the state-of-the-art quantitatively. It is suggested that the proposed approach outperforms others on the GTSDB database, considering generality, reliability and run time. Treating all traffic signs as one category, we have achieved 84.5% mAP and 97.81% recall.

A detection network proposed in [9] were trained and evaluated on Tsinghua-Tencent 100K and achieved 84% accuracy and 94% recall. Because different benchmark databases are used, we cannot compare with [9] equally. It should be noted that we trained a deep network using much lesser but more challenging images than [9]. Moreover, the proposed detection network can be trained and tested much faster. To further evaluate the generalization of our detector, we predicted the images randomly sampled from Tsinghua-Tencent 100K database and the qualitative results are also available online.

B. The Performance of Classification

The classification accuracy is 100% on the validation subset. And 99.66% accuracy is achieved on the evaluation subset. The classifier proposed here is compared with the state-of-the-art in Table V quantitatively.

The results suggest that the proposed approach is comparable to the sate-of-the-art. Besides, the structure designed here is fast to implement and train. The proposed classifier can predict a traffic sign image in 0.26 ms on a single GPU, which is up to real-time applications. It should be noted that 99.01% of the traffic signs get confidence probabilities more than 0.85 if they are classified correctly. As a result, we refuse to output the label of the candidate traffic sign whose confidence probability is less than 0.85 in Figure 2.

In order to demonstrate the advantage of the designed classifier, we make a self-comparison (see Table VI). The classifier with symmetrical kernel in Table VI uses 3×3 and 7×7 convolutions only. We remove the branch 1 (layer 6-1 and layer 7-1) and the branch 2 (layer 6-2 and layer 7-2)

Fig. 11. Examples of detection results. The results show that our detector is very robust, however, some typical negative samples are difficult to eliminate due to similar geometric shapes or contents. Some indication signs which are similar to the standard traffic signs such as the right-most detected sign in (a) and the middle detected sign in (b). Some objects which have circular or triangular shapes such as the wheel in (d). Traffic lights with red or blue color such as the false positive in (g). Some background regions which have triangular or rhombic shapes (boundaries) such as the two false positives on the left in (a). Moreover, very small traffic signs are difficult to detect such as the undetected traffic sign on the right which is called "speed limit 70" in (h).

TABLE IV
DETECTION RESULTS ON GTSDB [2], [14]

Method	Prohibitory (161)	Mandatory (49)	Danger (63)	Unique (92)	Time (s)
HOG+LDA+SVM [16]	100.00%	100.00%	99.91%	-	3.533
HOG+SVM [17]	100.00%	96.98%	100.00%	-	$0.4 \sim 1$
ROI+HOG+SVM [18]	99.98%	95.76%	98.72%	-	3.032
HOG+CNN [19]	-	97.62%	99.73%	-	$12 \sim 32$
ROI+HOG+SVM [3]	99.29%	96.74%	97.13%	-	0.162
ROI+Multi-task CNN [14]	99.99%	98.72%	98.34%	-	$0.366 \sim 0.450$
Ours	96% (154/161)	100% (49/49)	100% (63/63)	99% (91/92)	0.130

TABLE V
CLASSIFICATION RESULTS ON GTSRB

Method	Data Augmentation	Input Size	Parameters	Accuracy
Hinge Loss Trained CNN [7]	Yes	47×47	1.16M	99.65%
STN [31]	No	32×32	3.45M	99.59%
Committee of CNNs [5]	Yes	48×48	90M	99.46%
Multi-Scale CNN [6]	Yes	32×32	1.4M	99.17%
Random Forests (HOG 2) [37]	-	-	-	96.14%
LDA (HOG 2) [37]	-	-	-	95.68%
Human Performances [37]	\	\	\	98.84%
Ours	Yes	48×48	2.92M	99.66%

TABLE VI
SELF-COMPARISON OF DIFFERENT CNN STRUCTURES

Structure	Data Augmentation	Parameters	Accuracy
Symmetrical Kernel	Yes	3.42M	99.46%
Without Branch 1	Yes	2.82M	99.24%
Without Branch 2	Yes	2.79M	99.17%
More Hidden Neurons	Yes	10.03M	99.23%
Ours (final)	Yes	2.92M	99.66%

of the inception module in Table III in turn and evaluate the new structures. In addition, we increase the number of neurons in the hidden layer (layer 15) to 1024 and evaluate the new model. All the variant CNNs are trained using the Adam optimization algorithm. The results in Table VI indicate that the asymmetric kernels, the appropriate light-weighted hidden layer and feature fusion (inception module) can improve the classification accuracy of traffic sign objects.

C. The Necessity of Localization Refinement

Zhu *et al.* [21] have shown that improved localization can lead to better classification. In Section III-B, we have designed a localization refinement approach for specific categories of traffic signs. Here, we resize the refined bounding box and the ground truth bounding box to the fixed spatial extent of 48×48, and a refinement result is considered to be successful if the mean absolute error of refined coordinates is within 4 pixels. We successfully refine 95.2% of such traffic signs in the GTSDB database. For the shortage of images in the GTSDB database, we did a simulation experiment on the GTSRB database with 12630 evaluation images, analyzing the gap between the detection task and the classification task.

We continue to train our classifier with more shift augmentation on the GTSRB training subset to make the classifier more robust to imprecise localizations. Then we randomly shift the evaluation images horizontally and vertically within [$-\alpha$, α] pixels (obey uniform distribution). After that, we evaluate the accuracy of the classifier again. The results in Table VII show that the less precise the bounding box is, the less accurate the classifier is. That is to say, we should pay attention to the gap between detection and classification.

TABLE VII
THE BOUNDING BOX QUALITY AND THE CLASSIFICATION ACCURACY

α	4	6	8	12	14
Accuracy	99.28%	99.19%	99.09%	98.60%	97.40%

VI. CONCLUSIONS

In this paper, a complete pipeline of traffic sign recognition is proposed, which is superior to the majority of previous work, considering generality, reliability and run time. The detector and the classifier proposed here can be trained end-to-end and are capable of real-time applications. The proposed detector has been proved to be able to detect traffic signs beyond the limitations of colors and shapes. And the proposed classifier has been proved to be simple but very powerful.

However, the recall of traffic signs requires further improvement. According to the evaluation results, we believe that the performance will be boosted obviously, provided that more annotated data is used. The approach for the localization refinement of bounding box is designed for specific categories of traffic signs and is not robust enough, which will be further studied in our future work.

REFERENCES

[1] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, "Detection of traffic signs in real-world images: The German traffic sign detection benchmark," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.

[2] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2011, pp. 1453–1460.

[3] Y. Yang, H. Luo, H. Xu, and F. Wu, "Towards real-time traffic sign detection and classification," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2022–2031, Jul. 2016.

[4] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal, "On circular traffic sign detection and recognition," *Expert Syst. Appl.*, vol. 48, pp. 67–75, Apr. 2016.

[5] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3642–3649.

[6] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2011, pp. 2809–2813.

[7] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.

[8] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, pp. 758–766, Nov. 2016.

[9] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2110–2118.

[10] E. Peng, F. Chen, and X. Song, "Traffic sign detection with convolutional neural networks," in *Proc. Int. Conf. Cogn. Syst. Signal Process.* Singapore: Springer, 2016, pp. 214–224.

[11] H. Gómez-Moreno, S. Maldonado-Bascón, P. Gil-Jiménez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 4, pp. 917–930, Dec. 2010.

[12] A. Ruta, Y. Li, and X. Liu, "Real-time traffic sign recognition from video by class-specific discriminative features," *Pattern Recognit.*, vol. 43, no. 1, pp. 416–430, 2010.

[13] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.

[14] H. Luo, Y. Yang, B. Tong, F. Wu, and B. Fan, "Traffic sign recognition using a multi-task convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1100–1111, Apr. 2018.

[15] D. Wang, X. Hou, J. Xu, S. Yue, and C.-L. Liu, "Traffic sign detection using a cascade method with fast feature extraction and saliency test," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3290–3302, Dec. 2017.

[16] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, "A robust, coarse-to-fine traffic sign detection method," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–5.

[17] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu, "Traffic sign detection by roi extraction and histogram features-based recognition," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–8.

[18] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. Di Stefano, "A traffic sign detection pipeline based on interest region extraction," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–7.

[19] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2013, pp. 1–7.

[20] A. Shustanov and P. Yakimov, "CNN design for real-time traffic sign recognition," *Procedia Eng.*, vol. 201, pp. 718–725, Dec. 2017.

[21] Z. Zhu, J. Lu, R. R. Martin, and S. Hu, "An optimization approach for localization refinement of candidate traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3006–3016, Nov. 2017.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[23] A. G. Howard *et al.* (2017). "MobileNets: Efficient convolutional neural networks for mobile vision applications." [Online]. Available: https://arxiv.org/abs/1704.04861

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[25] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.

[26] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 779–788.

[27] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016.

[28] N. Barnes, A. Zelinsky, and L. S. Fletcher, "Real-time speed sign detection using the radial symmetry detector," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 2, pp. 322–332, Jul. 2008.

[29] S. Muller-Schneiders, C. Nunn, and M. Meuter, "Performance evaluation of a real time traffic sign recognition system," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2008, pp. 79–84.

[30] B. Hoferlin and K. Zimmermann, "Towards reliable traffic sign recognition," in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2009, pp. 324–329.

[31] T. L. Yuan. *GTSRB_Keras_STN*. Accessed: Nov. 1, 2017. [Online]. Available: https://github.com/hello2all/GTSRB_Keras_STN

[32] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3150–3158.

[33] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[34] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, 1981.

[35] M. Lin, Q. Chen, and S. Yan. (2013). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400

[36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[37] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.

[38] Tensorflow. *MobileNet_v1*. Accessed: Nov. 1, 2017. [Online]. Available: https://github.com/tensorflow/models/blob/master/research/slim/nets/mob% ilenet_v1.md

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[40] D. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

Jia Li was born in 1993. He received the B.S. degree in automation from Hefei University of Technology, Hefei, China, in 2016. He is currently pursuing the master's degree with the Department of Automation, University of Science and Technology of China. His research interests are in computer vision and deep learning.

Zengfu Wang (M'13) received the B.S. degree in electronic engineering from University of Science and Technology of China in 1982 and the Ph.D. degree in control engineering from Osaka University, Japan, in 1992. He is currently a Professor with the Institute of Intelligent Machines, Chinese Academy of Sciences, and the University of Science and Technology of China. He has authored over 200 journal articles and conference papers. His research interests include computer vision, human–computer interaction, and intelligent robots. He received the Best Paper Award from the ACM International Conference on Multimedia in 2009.