TRAFFIC SIGN RECOGNITION USING A MULTI- TASK CONVOLUTIONAL NEURAL NETWORK

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

AYYAPPADAS CHANDRAN

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

TRAFFIC SIGN RECOGNITION USING A MULTI- TASK CONVOLUTIONAL NEURAL NETWORK

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

AYYAPPADAS CHANDRAN

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Traffic Sign Recognition Using A Multi- task Convolutional Neural Network** *submitted by* **Mr. AYYAPPADAS CHANDRAN**, *Reg. No.* **MAC15CS019** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by him under our supervision and guidance.*

....	....	....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the God Almighty for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Traffic sign recognition plays an important role in Driver Assistance Systems and Automated Driving. Although traffic sign recognition has been studied for many years, most existing works are focused on the symbol-based traffic signs. In the proposed method all categories including both symbol-based and text-based ones are processed and a new multi-task CNN which treats the tasks of ROIs(Region Of Interest) refinement and classification jointly is proposed. To address the problem of relatively small amount of labelled traffic signs, two kinds of data acquisition methods street view images and synthetic images are combined to obtain large number of labelled samples with low cost. The three stages of recognition system are ROIs extraction, ROIs refinement and classification, and post processing. Experimental results have demonstrated the effectiveness of the proposed system.

Contents

List of Figures

List of Abbreviation

GTSRB	German Traffic Sign Recognition Benchmark
GTSDB	German Traffic Sign Detection Benchmark
ROI	Region Of Interest
CNN	Convolutional Neural Network
MSER	Maximally Stable Extremal Regions
SVM	Support Vector Machine
HOG	Histogram of Oriented Gradients
ACF	Aggregate Channel Features
ICF	Integral Channel Features
LDA	Linear Discriminant Analysis

Introduction

Traffic sign recognition plays an important role in Driver Assistance Systems and Automated Driving. However, this task is not easy for a computer because of the large variations in visual appearance of traffic sign images due to partial occlusion, different viewpoints, illuminations and weather conditions, etc. To recognize traffic signs in an image, most popular methods include two steps: Detection and Classification. There are a lot of researchers working on this challenging task with the already popular or specially designed vision algorithms. However, it is not easy to compare these methods since there did not exist a public available data set until the release of the German Traffic Sign Recognition Benchmark(GTSRB) [1] and German Traffic Sign Detection Benchmark(GTSDB) [2] in 2011 and 2013 respectively. Since then, researchers can evaluate and compare their algorithms on the same benchmarks.

Nevertheless, there still exist some defects in the GTSDB and GTSRB: 1) they include only three categories of symbol based traffic signs with regular shape and color which are relatively easy to detect and classify, while text-based traffic signs are more challenging; 2) the GTSDB only includes static images, but in real scenarios, continuous video captured by an in-vehicle camera is useful for detection and classification; 3) the final task of traffic sign recognition is to know the existing signs in a scene, but the two benchmarks separate it into two independent tasks with different datasets.

To alleviate these problems, we propose a new system to recognize existing traffic signs from video input and evaluate its performance on a new challenging data set with the following features: 1) it contains both the symbol-based and text-based traffic signs, up to seven categories, which is contrasted to the previous three symbol-based categories; 2) instead of a static image, each sample in the data set is a short video of 5 to 20 low quality frames which is captured from an in-vehicle camera. Some examples in the data set are shown in Fig. 1.1. It can be seen that symbol-based traffic signs have the same appearance with discrepancies in viewpoint, illumination, blur, background and so on, while text-based signs may have very different appearances even within the same class.

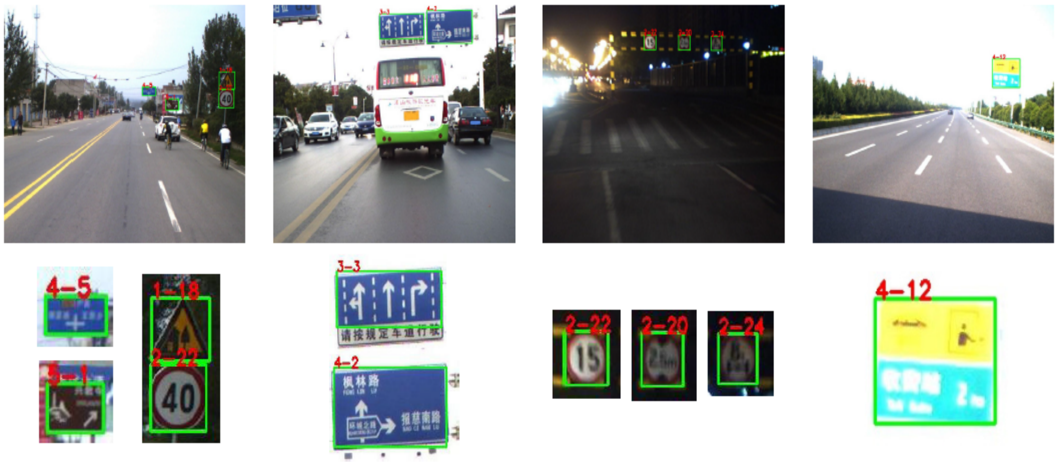


Fig. 1.1: Examples in the challenging data set.
The first row shows the raw images; the second row shows the cropped traffic signs. The target traffic signs (green bounding boxes) and their classes (text near the boxes) are also shown in the images. The traffic signs, such as 4-5, 5-1, 3-3, 4-2, 4-12, are text-based, and 1-18, 2-22, 2-20, 2-24 are symbol-based.

Our traffic sign recognition system consists of three stages: traffic sign regions of interest (ROIs) extraction, ROIs refinement and classification, and post-processing. First, for each frame in the video, traffic sign ROIs are detected with Maximally Stable Extremal Regions (MSERs) on multi-channel images. Then, to refine and classify the ROIs, a multi-task Convolutional Neural Network (CNN) is proposed. Specifically, the ROIs are first fed to a binary classification layer, and only the positive ones are further classified with a deep multiclass classification network. The network is trained end-to-end with a large number of data, which consists of training data, synthetic signs and images labeled from street view. Finally, recognition results from each frame are fused to get the final results of the video. Such a system pipeline is illustrated in Fig. 1.2.

The main contributions of this paper are as follows: 1) while many existing works are focused on symbol-based traffic signs, we process all categories, including both symbol-based and text-based ones; 2) a new multi-task CNN which treats the tasks of ROIs refinement and classification jointly is proposed 3) to address the problem of relatively small amount of labeled

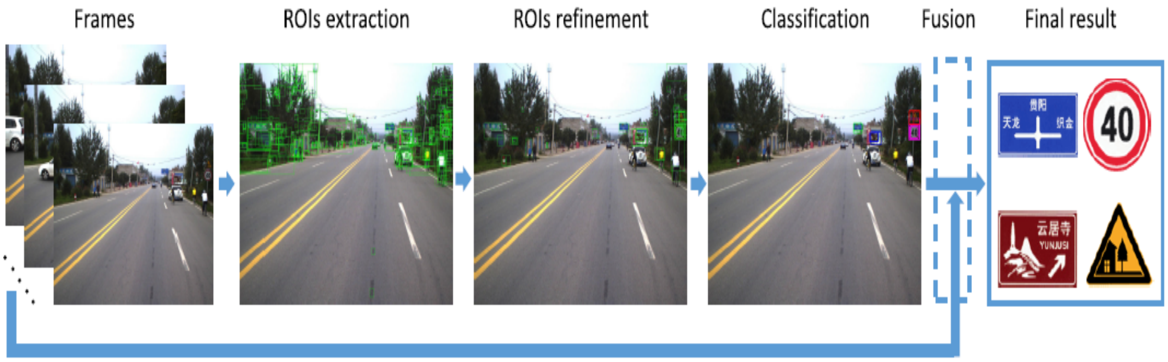


Fig. 1.2: Pipeline of the proposed traffic sign recognition system.

traffic signs, two kinds of data acquisition methods, street view images and synthetic images, are combined to obtain large number of labeled samples with low cost; 4) our system achieves the best result in a newly released, challenging data set.

Related works

Generally, traffic sign recognition contains two parts: detection and classification. The purpose of detection is to find the locations and sizes of the existing traffic signs in an image, and the task of classification is to assign a class label to each detected traffic sign. Related works on these two parts are reviewed respectively in this section.

2.1 Traffic sign detection

The images obtained by the camera are often of poor quality due to the sophisticated environment conditions. Low-level image pre-processing can be used to enhance the traffic sign regions of the captured images, which makes it easier for subsequent tasks. The most common way is transforming images into a new color space where the signs are more distinct. Many color spaces have been used, such as HSI, improved HLS and normalized color space. Another pre-processing method is using machine learning to learn the color space mapping from data. Reference [3] proposed a color probability model which can enhance the main color of the signs while suppressing the background regions. SVM classifier was trained to map each pixel in color images to a gray value which has high response in sign regions.

In the early stage of object detection, it was popular to use threshold based methods. In [4], different threshold based segmentation methods were compared. This kind of methods are not robust in complex environment with unpredictable lighting conditions.

Recently, machine learning based object detection is becoming dominant in the research community. On the traffic sign detection, there are sliding window based methods and region of interest (ROI) based methods.

Sliding window based methods such as Integral Channel Features(ICF), Aggregate Channel Features (ACF) were used in [5] and [6] to detect traffic signs. Other methods like Adaboost with the enhanced channel features , Adaboost with the Haar-like features, SVM with color

HOG were reported too. These approaches are time-consuming since they need to construct a multi-scale pyramid. What is worse, it is difficult to determine the sliding window size and its aspect ratio.

Another approach is to first extract regions of interest (ROIs) and then filter out non-object ROIs with a classifier. Compared to the sliding window based methods, it reduces the computational time and does not need to tune the parameters of sliding window. An important consideration of this method is the recall rate of target objects among the extracted ROIs. It is expected to have as high recall rate as possible while keeping the number of ROIs as low as possible. Given that traffic signs are designed with a large part of uniform region, MSERs have been proved to be very effective in extracting such ROIs. In [7] and [8], a coarse sliding window method was used to extract ROIs. Template matching was also used for ROIs extraction. Filtering out non-sign objects from ROIs can be treated as a classification task. SVM classifier with HOG features is the most popular framework due to its excellent performance. Some other methods like Convolutional Neural Network(CNN), Extreme Learning Machine were also used.

2.2 Traffic sign classification

Traditional methods for classification include feature extraction and classifier training. Some combinations reported in literature include a cascade of SVM classifiers with HOG features, K-d trees and Random Forests with Distance Transforms and HOG features, MLP(Multi-Layer Perceptron) with radial histogram features, ANN (Artificial Neutral Network) with RIBP (Rotation Invariant Binary Pattern) based features, SVM with LIPID (local image permutation interval descriptor), etc. In [9], dense SIFT features, HOG features and LBP features were first extracted, then they were encoded through locality-constraint linear coding (LLC) and the resulting codes were pooled by spatial pyramid pooling(SPM). The three different feature representations were concatenated as the final features of a traffic sign, and a linear SVM was used as the classifier. In general, it is very laborious and difficult to design a good feature.

Convolutional Neural Network(CNN) which can be trained without the need of hand-designed features is popular nowadays. Multi-column CNNs which train multiple CNNs with different weight initialization or data preprocessing, were proposed to classify traffic signs. Although CNN has shown its excellent performance in image classification, how to design a good network architecture and train a workable model are still challenging tasks. To handle the geometry variations of traffic signs, data augmentation was used to enlarge the training data set. Raw color image and four channels used to extract MSERs. (a) Original color image. (b) Gray and normalized R,G,B channel.

Another method is to eliminate the geometry variations. The traffic signs were first classified into several super classes, for each of which perspective adjustment was performed with a specially designed method. Then, the adjusted signs were classified into their detailed classes. Recently, Spatial Transformer Network (SPN) was proposed in [10], which can explicitly learn geometry parameters of transformation, and be robust to the geometry variations of input images.

Proposed method

3.1 Traffic sign recognition

In this section, we describe our method for recognizing traffic signs from video. First, traffic sign ROIs from each frame of the video are extracted using MSER. Then, the ROIs are refined and recognized with a multi-task CNN. Finally, the outputs of all frames are fused to get the final recognition result in the post-processing stage.

3.1.1 Traffic sign ROIs Extraction

Since the traffic signs in real traffic scene are of large differences in colors, shapes and sizes, it needs a lot of tricks to detect them with sliding window based method. Fortunately, there are large portion of uniform areas within traffic signs, which can be easily detected by MSER. For this reason, we use MSER to extract traffic sign ROIs from each frame of the video. Since the main colors of traffic sign are different in different type of signs, multi-channel images are used to extract MSERs to increase the recall rate of the extracted ROIs. Four channels used in this paper include gray and normalized RGB channels computed by

$$\text{norm}_r = R/(R+G+B)$$
$$\text{norm}_g = G/(R+G+B)$$
$$\text{norm}_b = B/(R+G+B)$$

When extracting MSERs, we use prior knowledge about the size and shape of traffic sign to eliminate a few number of non applicable ROIs. Specifically, the size of a traffic sign ROI (represented by a bounding box) should lie within a range. The shape of a ROI can be represented by the length-width ratio of the bounding box, and this value should also have a restriction. All the parameters can be easily determined from the statistics results on training data. Besides the ROIs around the traffic sign, there are many ROIs in the regions such as buildings and trees, which appear frequently in a traffic scene.

3.1.2 Multi-task CNN for ROIs refinement and classification

After the traffic sign ROIs extraction, traffic signs and a large number of backgrounds are obtained. The tasks of this stage are to filter out backgrounds and determine the detailed classes of the remaining ROIs, namely ROIs refinement and classification, respectively. Traditionally, most related works tackled the two tasks separately. For ROIs refinement, the widely used method is using SVM classifier with HOG features. For the task of classification, CNN is the mainstream method, which has been proved to be an excellent model in computer vision. It can be trained end-to-end from the image data, and no manually designed features are needed any more. In this paper, we propose a new CNN architecture which unifies the two tasks. We call this architecture as multitask CNN. There are two important issues in the CNN-based method, structure of the network and large amount of training data. In this subsection we will describe the structure of proposed CNN, and the method for acquiring enough training data will be introduced in the next section.

There are two decision layers in the proposed multi-task CNN. One is called binary classification layer for distinguishing backgrounds and traffic signs, and the other is called multiclass traffic sign classification layer. They correspond to the tasks of ROI refinement and classification respectively in the traditional methods. Here, the binary classification layer aims to fast eliminate most background ROIs and allows some hard backgrounds to pass, which will be removed by the multiclass classification layer based on deeper features. During the training and testing stages, all ROIs are first fed to the binary classification layer, and only the positive ROIs are fed to the next part of the network to obtain the detailed classes. In the training stage, loss from both decision layers are used to jointly optimize the network.

The basic structure of the multi-task CNN is shown in Fig. 3.1. The conv(k,m) means a convolutional layer with kernel size $k \times k$, and output channel number m. Appropriate padding precedes all convolutional operations so as to keep the width and height of input channel, and the stride is 1 in all operations. The relu denotes the rectified linear unit (ReLU) layer. The maxpooling(k) means max pooling layer with kernel size $k \times k$ and stride k.

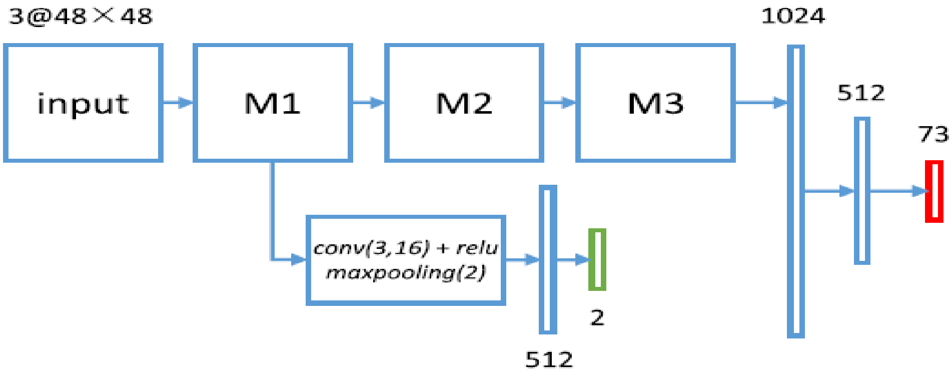
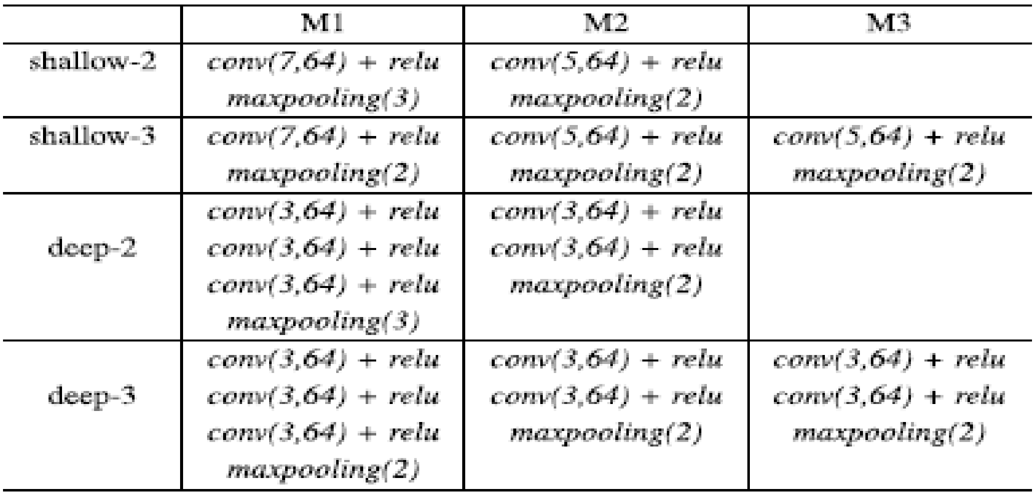


Fig. 3.1: The structure of the multi-task CNN.
There are two decision layers, one for binary classification (green layer) and another for multi-class classification (red layer).



	M1	M2	M3
shallow-2	*conv(7,64) + relu* *maxpooling(3)*	*conv(5,64) + relu* *maxpooling(2)*	
shallow-3	*conv(7,64) + relu* *maxpooling(2)*	*conv(5,64) + relu* *maxpooling(2)*	*conv(5,64) + relu* *maxpooling(2)*
deep-2	*conv(3,64) + relu* *conv(3,64) + relu* *conv(3,64) + relu* *maxpooling(3)*	*conv(3,64) + relu* *conv(3,64) + relu* *maxpooling(2)*	
deep-3	*conv(3,64) + relu* *conv(3,64) + relu* *conv(3,64) + relu* *maxpooling(2)*	*conv(3,64) + relu* *conv(3,64) + relu* *maxpooling(2)*	*conv(3,64) + relu* *conv(3,64) + relu* *maxpooling(2)*

Fig. 3.2: Detailed structure of 4 CNN

To design a good network, it is important to consider the depth of the network. In this paper, we define and compare four network structures with different depths and convolutional kernel sizes. They have the similar basic structure as shown in Fig. 3.1. Specifically, the input of network is color image with size of 48×48. The node number of two decision layers are 2 and 73, which represent background and traffic sign in the first case and the number of sign classes and an additional background class in the second case. ReLU layer is added after each full connected layer except the final decision layers. Dropout layer with a probability of 0.5 is added after the two full connected layers which connect from pooling layers. The detailed structures of the four models are listed in Fig 3.2. In the shallow models, all convolutional layers are with big filters, while the deep models split big filters into a few number of small filters with fixed size of 3×3 . Each deep or shallow model includes 2 or 3 maxpooling layers, so there are four models in total.

3.1.3 Post processing

Each extracted traffic sign ROI is fed to the above multi-task CNN to get the classification result. This operation is applied to each frame of a test video. In nearby frames of the video, the recognition results may be slightly different due to their different appearance in different frames. For this reason, it is necessary and important to fuse results from all frames to get the final recognition result in a short video.

For the classified positives, we set a threshold $thresh_p$ to remove candidates with low confidence. For all the remaining candidate signs, the frequency of each class in all frames is calculated, and a traffic sign class whose frequency is larger than a frequency threshold $thresh_f$ is assigned as an existing sign class in the video. The two thresholds can be determined by grid search on validation dataset.

3.2 Data preparation for training CNN

In our system, both the detection stage and classification stage rely on the CNN model. Due to the complexity of CNN structure, it is required to have as much labeled data as possible to train a reliable CNN model. For this purpose, we propose two different ways to acquire additional data for training. One is to use the street view images and the other is to use synthetic traffic signs.

It is worth noting that the two methods were separately used in some related works. Google Street View images were used to help the development of vision-based driver assistance systems. Generating synthetic data from standard traffic signs was used to detect and classify traffic signs in some works. In this paper, the aim is to recognize all categories of traffic signs, and it is more challenging than the previous works. Both types of data are used, and the contribution of each type of data is evaluated in our experiments

3.2.1 Collect training data by street view images

Classically, to collect training data for a traffic sign recognition system, one needs a car with a car-mounted camera driving around the streets. This method is time consuming and expensive. Recently, many web map service providers offer street view for many countries. Street view images contain a large number of traffic signs in real traffic scenes, and they can be obtained with very low cost. As a result, we propose to capture street view images as additional training data for our CNN models.

As described in the first section, text-based traffic signs have larger within-class variations. Therefore, we paid more attention to the text-based signs when exploring the street view map. After the street view images were acquired, we annotated these images with information such as locations, sizes and the classes of the existed signs.

For the annotated street view images and training data, ROIs are extracted first. Then, the Intersection over Union (IoU) score calculated by

$$IoU = \left(\frac{area(gt \cup dt)}{area(gt \cap dt)}\right) \qquad \text{(Equ:3.1)}$$

IoU is used to assign a label to each ROI, where gt,dt are ground truth and detected box respectively. For each ROI, its IoU scores to all traffic sign ground truths are calculated, and the maximal score and corresponding class are recorded. A ROI with IoU score larger than 0.5 is assigned as a sample of corresponding class, while a ROI with score less than 0.2 is labeled as a sample of background class. ROIs in the gray zone (with IoU scores between 0.2 and 0.5) are ignored. We set a gray zone to decrease the probability of a weak detected sign (a traffic sign with low IoU score) being classified as a background. By using this method, most ROIs are assigned to background class due to the fact that the signs only occupy a small part of image. To balance the sample numbers between background class and other traffic signs, we use different sampling and processing method for them. For traffic sign ROIs, each one is copied 25 times, while each background ROI is copied once. To further increase the number of traffic sign samples, the data augmentation method is used on ground-truth traffic signs. For each labeled traffic sign, random translation, rotation and scale transformation are applied.

3.2.2 Collect training data by synthetic traffic sign images

In the street view images, there are also a lot of symbol based traffic signs. However, their distribution is not uniform. For example, speed limit signs are very common in real traffic scene, while the signs such as weight limit signs only exist in some special scenes. For the same reason, there are many subclass signs with different speed values in speed limit signs, and some signs with very low or high speed values are also not common in many scenes. To acquire more data with low cost and balance data quantity in each class at the same time, we also synthesize traffic sign images from standard sign templates. The pipeline of how to synthesize data is shown in Fig. 3.3

First, we get all standard signs that need to be classified. The standard signs are manually processed to add a mask channel which is important to synthesize the sign images with background. To simulate the viewpoint change in the real scene, random planar affine transformation is applied to standard signs. A planar affine transformation has the matrix representation of equation (3) with six parameters.

$$\begin{pmatrix} x^{'} \\ y^{'} \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & d_x \\ a_{21} & a_{22} & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \text{(Equ:3.2)}$$

or in the block form

$$x' = Hx = \begin{pmatrix} A & d \\ 0' & 1 \end{pmatrix} x \qquad \text{(Equ:3.3)}$$

where A is a $2{\times}2$ non-singular matrix.However, the elements in A are the mixed results of geometry parameters such as rotation and scale, making it difficult to simulate. Fortunately, the affine matrix A can be decomposed as

$$A = \lambda R(\theta) R(\phi) \begin{pmatrix} t & 0 \\ 0' & 1 \end{pmatrix} R(-\phi) \qquad \text{(Equ:3.4)}$$

where R(.) denotes a rotation matrix.We can control the six independent parameters, dx, dy, λ, t,θ,ϕ to simulate geometry transformation.

The bounding boxes of detected traffic signs in real scenarios include both the sign and background, especially for those non-rectangle signs. To model this property, we add background to the transformed sign generated in the previous stage. The background images are collected from real traffic scene without any signs. First, a patch of the same size as the sign is randomly cropped from the background image set. Then the sign image is added with the background image patch as following:

$$I_{out} = mask \odot I_{sign} + (1 - mask) \odot I_{background} \qquad \text{(Equ:3.5)}$$

where $I_{out}, I_{sign}, I_{background}$ are output image, input sign image and background image patch respectively, and denotes element-wise multiplication. The mask is the mask channel matrix of the sign image, which has binary values. Given the variations in the time and weather conditions, we also simulate the lighting variation. To simplify this procedure, we only fuse the images with different lighting images as follow:

$$I_{out} = weight \times I_{sign} + (1 - weight) \times I_{light} \qquad \text{(Equ:3.6)}$$

where I_{light} denotes the lighting image, and weight balance two images. Similar to background image patch, the lighting image is also randomly cropped from lighting image set which contains images with varying luminance.

Finally, the synthetic images are blurred with Gaussian kernel of random size. Some synthetic images are shown in Fig 3.3. It can be seen that they have different appearances, which are crucial to train a network to recognize traffic signs with many variations.

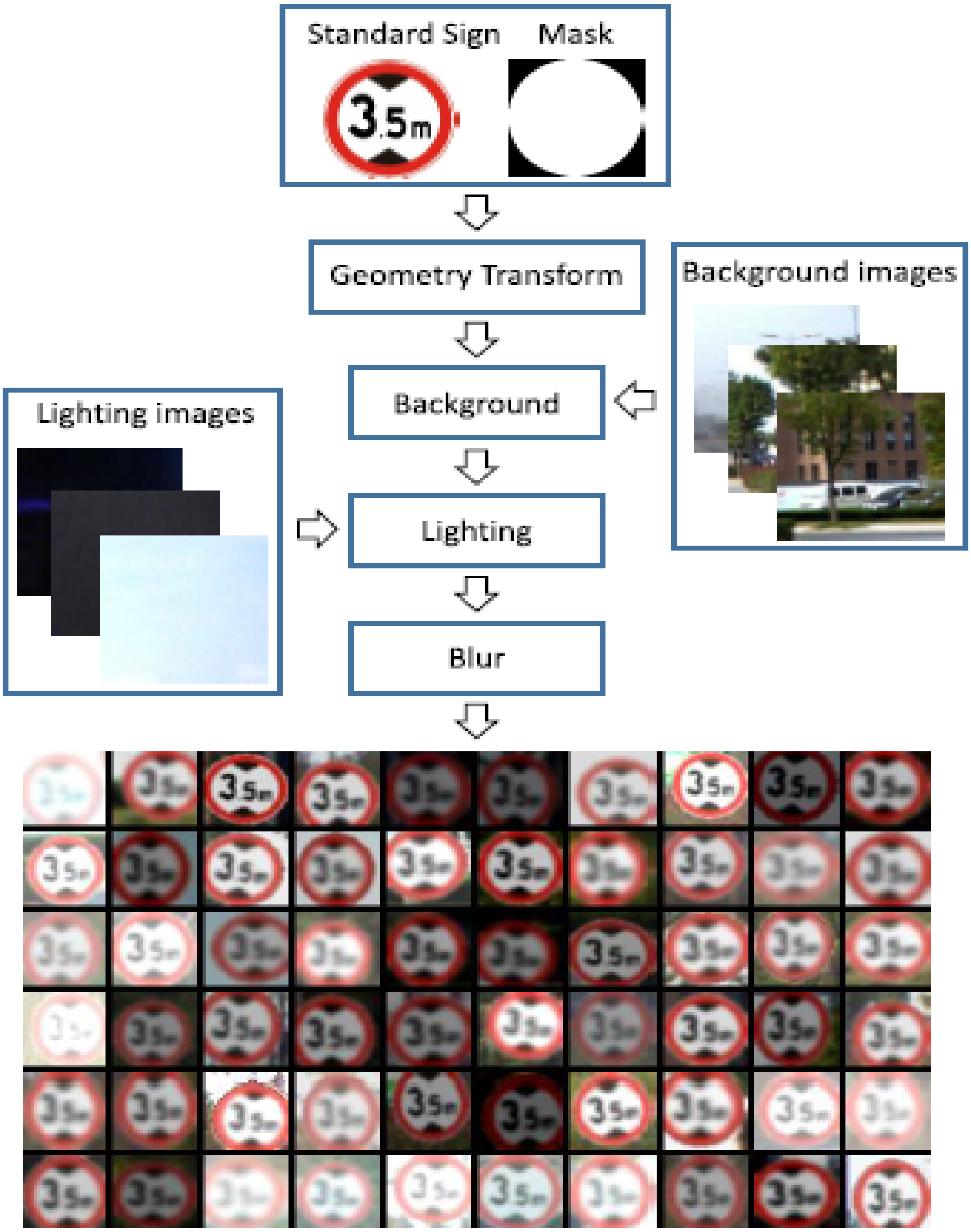


Fig. 3.3: Pipeline of synthesizing traffic signs.

3.3 Performance analysis

In this section, we will evaluate the proposed traffic sign recognition system on a new challenging dataset. First we give a brief description of the data set. Then the parameters of each module in this system are evaluated, along with some comparison experiments. Finally, we report the result we got and the results obtained by our competitors.

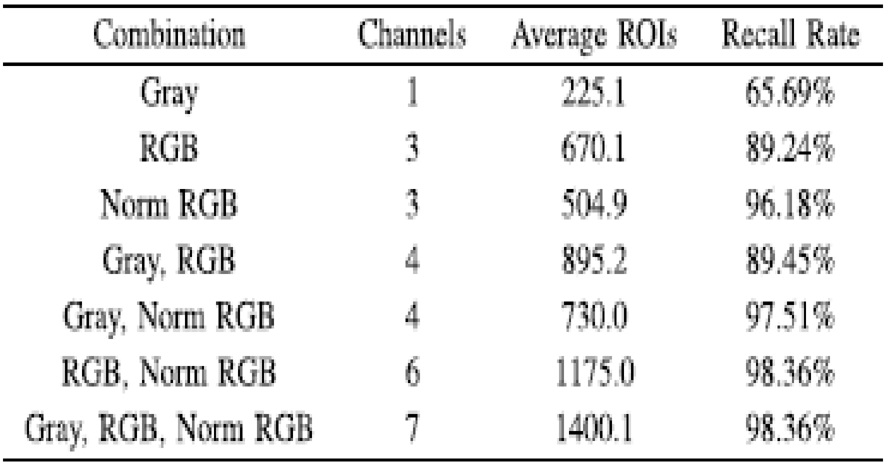
3.4 Training data

Besides the provided training data, we use the street view images and synthetic signs additionally to train our CNN models. There are totally 887 frames in the training videos. The number of street view images we captured is 2042. Data augmentation is used on both of them. The translation parameters are G[w/8,w/8], G[h/8, h/8], where G[a, b] denotes $\frac{b+a}{2} + \frac{b-a}{5}$ $N(0, 1)$, and w, h are the width and height of one sign. The rotation and scale transformation parameters are G[/20, /20], U[0.9, 1.1] respectively, where U[a, b] denotes the uniform distribution. In the process of data augmentation, if the IoU score between the new generated sign and ground truth is less than 0.5, we will resample with another random parameters. For each traffic sign labeled in the data set, it is augmented to the number of 200.

The third part of the training data is the synthetic image data. As described in previous section, there are six parameters, dx, dy,λ, t, θ, ϕ to control the geometry transform. The standard signs are first resized to 48 48, and the six parameters are randomly sampled from G[6, 6], G[6, 6], U[0.9, 1.1], U[0.9, 1.1]/, G[/20, /20], U[/36, /36] respectively. The weight used in (7) is randomly sampled from U[0.4, 1.0]. The size of Gaussian kernel for blurring the synthetic image is randomly selected from the set $3 \times 3, 5 \times 5, 7 \times 7, 9 \times 9$. For each standard traffic sign, we synthesize 5000 images. Millions of training samples (including traffic signs and backgrounds) are obtained from these three types of data source. All images are resized to 4848 and randomly shuffled to train the CNN.

3.4.1 Performance of traffic sign ROIs extraction

In this subsection, we investigate the performance of combining different color channels, including gray channel, RGB channel, and normalized RGB channel. For each combination, the recall rate, average number of ROIs are recorded. Recall rate is the ratio between the number of hit signs and all signs in all frames. IoU score is used to test whether a sign is hit by ROIs or not. A sign is considered as a hit sign if there is at least one ROI with IoU score not less than 0.5. The results of different combinations are shown in Fig 3.4.

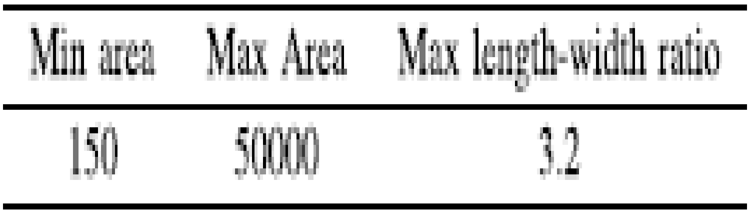


Combination	Channels	Average ROIs	Recall Rate
Gray	1	225.1	65.69%
RGB	3	670.1	89.24%
Norm RGB	3	504.9	96.18%
Gray, RGB	4	895.2	89.45%
Gray, Norm RGB	4	730.0	97.51%
RGB, Norm RGB	6	1175.0	98.36%
Gray, RGB, Norm RGB	7	1400.1	98.36%

Fig. 3.4: Combinations of different channels

From the table, we can see that normalized RGB channels are better than raw RGB channels. When combining more channels, higher recall rate will be obtained. However, it also needs more time and generates more ROIs. To balance the recall rate and the number of ROIs, the combination of gray and normalized RGB channels is finally used in this paper.

During the ROIs extraction, three parameters, the aspect ratio, maximum area and minimum area of the extracted region, are used to remove a few number of ROIs. These parameters are determined on the training data set. The detailed parameters are given in Fig 3.5.



Min area	Max Area	Max length-width ratio
150	50000	3.2

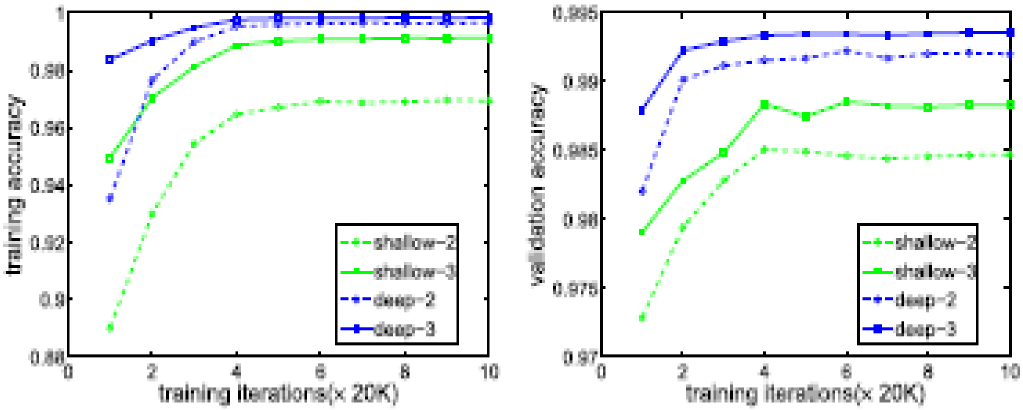
Fig. 3.5: Prior knowledge about traffic signs

3.4.2 Network structure

Four different network structures are compared in this work. To test their performance, we train the four networks with same data and compare the results on the validation data set. Since the depth of networks mainly influence the performance of multi-class classification, for simplicity, here we only evaluate the performance of different network structures with the simplified CNN which contains only multi-class classification layer.

Caffe and stochastic gradient descent(SGD) algorithm are used to train the networks. The mean of all pixels in train images is computed before the training. During the training, mean subtraction is used to process each pixel. Xavier initialization method is used to initialize all weights in the network, and all biases are initialized to 0. The momentum and weight decay are 0.9 and 0.0005 respectively. The initial learning rate of deep and shallow networks are 0.005 and 0.001 respectively, and these values are decreased to 0.1 times for each 50000 iterations. The batch size is set to 256, and total number of training iterations is 200000 for all networks. It needs about 1 day to train a network on a Titan Black GPU.

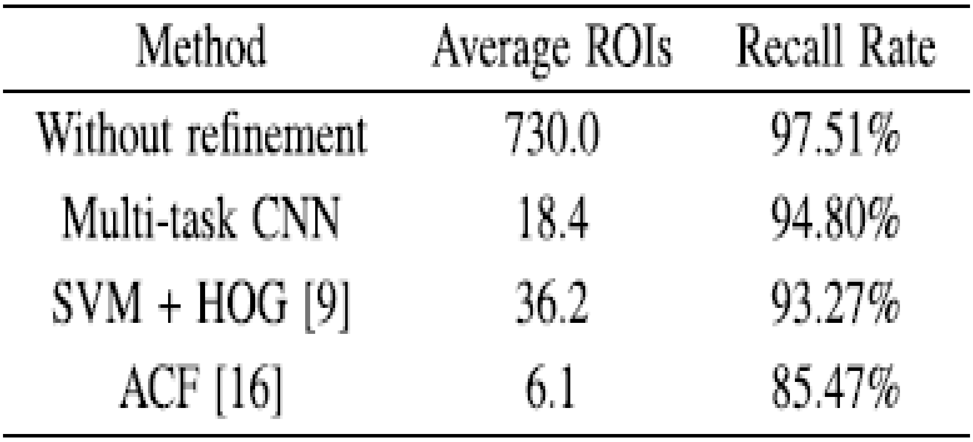
The training accuracy is shown in Fig. 3.6a. From this figure, we can see that the deep models are better than the shallow ones. To verify the generalization ability of the networks, we test all ROIs extracted from the validation data set. The ROIs with IoU score larger than 0.5 are assigned to corresponding sign classes, while others are assigned to background classes. The validation accuracy is shown in Fig. 3.6b. It shows the similar result as on the training data set.The deep model with 3 maxpooling layers (deep-3) is selected as the final network architecture due to its superior performance. In the following subsections, all experiments are



(a) (b)

Fig. 3.6: Training and validation accuracy of four different CNNs

conducted on the deep-3 with same learning parameters used in this subsection.



Method	Average ROIs	Recall Rate
Without refinement	730.0	97.51%
Multi-task CNN	18.4	94.80%
SVM + HOG [9]	36.2	93.27%
ACF [16]	6.1	85.47%

Fig. 3.7: Average number of ROIs and recall rate of the three methods

3.4.3 Performance of multi-task CNN

In this subsection, we will show the result of the proposed method. For our multi-task CNN, the loss weight between the binary and multi-class classification layer is a super parameter. We tried parameters from 2/1, 1/1, 1/2, 1/5, 1/10, 1/20, 1/50, and found that 1/20 was the best under the evaluation criterion of recognition score. The extracted ROIs can also be classified directly with a simple classification CNN. So the multi-task CNN is compared with a single-task CNN, which is the proposed multi-task CNN without binary classification part.

The proposed method is also compared with two others. One [3] is similar to ours, in which ROI based detection framework is used too. MSERs have been proved to be very effective in extracting ROIs for traffic signs, and are used in both methods. However, in the new dataset, there are more categories of traffic signs and dominate colors than the dataset used in [3], so more image channels are used to extract ROIs. Our method approaches the two tasks simultaneously in a multi-task CNN, while in [3], HOG+SVM was used to refine the ROIs, and the target ROIs were assigned to their detailed classes with a CNN. Given that the CNN structure proposed in [3] may be not deep enough for these complicated traffic signs in the new data set, the remaining ROIs are also classified by the proposed single-task CNN.

The other method compared in this paper is a state of the art sliding window based method. In [6], Integral Channel Features (ICF) detection framework achieved near-perfect detection result on GTSDB. The ICF and ACF(Aggregate Channel Features, an enhanced version of ICF) were also used in the detection of U.S. traffic signs and showed better result than previous works [5]. We choose to compare our method with ACF. The basic parameters are same as those in [6]. The depth of the weak learner is 2, and the classifier is trained in four stages with increasing number of weak learners(50, 100, 200 and 400). Since the traffic signs have different shapes and sizes in the new dataset, 7 ACF models with different sliding window sizes were trained for different categories of traffic signs in Fig. 7 ($a : 24 \times 24; b : 24 \times 24; c1 - c2, c4 - c12, d6 - d7 : 24 \times 24; c3, d1 - d5 : 24 \times 40; d8 - d12, e : 24 \times 24 and 24 \times 36; f, g : 16 \times 48$).During the multi-scale detection,8 down-sampled octaves are used, and each octave includes 8 scales. Given that ACF is only a detection framework, the detected traffic signs are classified by the proposed single task CNN.

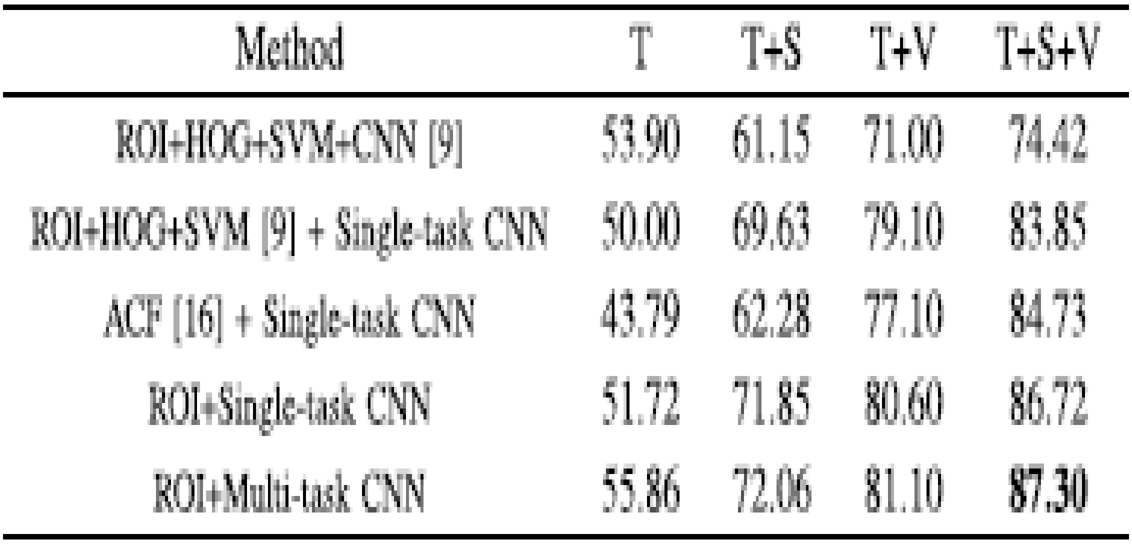
Our multi-task CNN and [3] both have ROI extraction and refinement parts. The ACF detects the traffic signs from full images directly. All the three methods obtain the target traffic signs with some false positives and false negatives. Fig 3.7 shows the average number of ROIs and recall rate of the three methods. Our method and [3] achieve higher recall rate than ACF

. The average number of ROIs of ACF method is less than two others. We can know from the table that the recall rate of ROI based method is higher than sliding window based method in this dataset including many different shapes and sizes of traffic signs.

After the ROI refinement or detection stage, the remaining ROIs are classified into their detailed classes. As mentioned before, the ROI+Multi-task CNN will be compared with ROI+single-task CNN method. Both methods classify the ROIs with CNN, however, the former one uses a multi-task CNN which refines and classifies the ROIs simultaneously, while the latter one classifies all ROIs directly with a CNN. In [3], the remaining ROIs were classified with a shallow CNN model. In this paper, these remaining ROIs from [3] are also classified by the proposed single-task CNN. To compare the final result of ACF with others, the detected traffic signs are classified by our single-task CNN. The results from each frame are fused to get the final recognition result. The post-processing method are the same for all methods.

The rightmost column of Fig 3.8 shows the final results of our multi-task and single-task CNN, and two other methods. Our proposed multi-task CNN obtains the best result. The single-task CNN obtains a little worse result. It indicates the effectiveness of the multi-task CNN, where the multiclass classification layer only focuses on the classification of the hard background ROIs and true traffic signs. However, the single-task CNN needs to distinguish a large number of background ROIs and true traffic signs, which makes it difficult to train a good classification model for traffic signs. The ACF + single-task CNN obtains worse result than ROI + single-task CNN, and it is mainly due to the relatively low recall rate of ACF model. The method in [3] does not work well in this dataset, and it is because the CNN proposed in [3] is not suitable for this hard dataset. When the shallow CNN is replaced by the single-task CNN, the result is much better, but still worse than the multi-task CNN. The multitask CNN can handle the ROI refinement and classification in a uniform framework and be trained end-to-end without the need of hand-designed features, while [3] and [6] split the ROIs refinement or detection and classification in two stages with different methods. These results indicate the effectiveness of the multi-task CNN.

With the proposed method, in the second stage competition, we got the first place among all six teams with the recognition score of 86.75%. The scores of the second and third place teams are 83.47% and 82.35%, respectively. CNN model has became the de facto method for image recognition, and all top 3 teams used it. Additional data were also used in all teams. We owe the good performance of our system to the specially designed network structure and carefully prepared training data. We try to include their methods for the completeness of this paper, however, there are no public available papers/technical reports about their methods. So, here we only give a coarse qualitative comparison. We got their methods information by the on-site presentation.

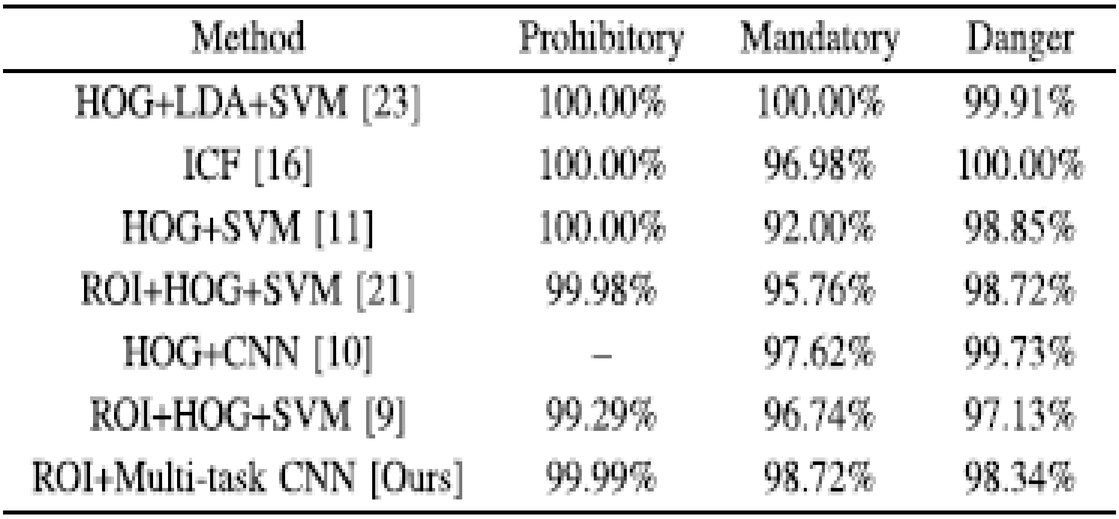


Method	T	T+S	T+V	T+S+V
ROI+HOG+SVM+CNN [9]	53.90	61.15	71.00	74.42
ROI+HOG+SVM [9] + Single-task CNN	50.00	69.63	79.10	83.85
ACF [16] + Single-task CNN	43.79	62.28	77.10	84.73
ROI+Single-task CNN	51.72	71.85	80.60	86.72
ROI+Multi-task CNN	55.86	72.06	81.10	**87.30**

Fig. 3.8: Recognition results on validation data set by using different training data

3.4.4 Performance on GTSDB

In this subsection, we test our method on the GTSDB. This data set contains 600 images for training and 300 images for testing. The task is to detect three categories of traffic signs, Prohibitory, Mandatory and Danger signs. First, the MSERs are extracted as ROIs. The recall rates of three categories are 161/161, 49/49 and 62/63, respectively. To detect the three categories of traffic signs, the proposed multi-task CNN is used. The only difference is the node number of the multiclass classification layer, which is 4 here (i.e., 3 categories of traffic signs and an additional background class). The given training data is first augmented with geometry, lighting and blur changing, and then used to train the network. By using the trained model, we can detect the three categories of traffic signs with their corresponding probability. Fig 3.9 show the area under the precision-recall curve (AUC) of our method and some state-of-the-art methods on GTSDB.



Method	Prohibitory	Mandatory	Danger
HOG+LDA+SVM [23]	100.00%	100.00%	99.91%
ICF [16]	100.00%	96.98%	100.00%
HOG+SVM [11]	100.00%	92.00%	98.85%
ROI+HOG+SVM [21]	99.98%	95.76%	98.72%
HOG+CNN [10]	–	97.62%	99.73%
ROI+HOG+SVM [9]	99.29%	96.74%	97.13%
ROI+Multi-task CNN [Ours]	99.99%	98.72%	98.34%

Fig. 3.9: Detection results on GTSDB for 0.5 overlap

3.4.5 Effect of data

To evaluate the contribution of the additional data used in this paper, we conduct experiments without or with only one part of the additional data and compare the recognition results. Four different combinations of data are compared, including given training data, given training + synthetic data, given training + street view data and given training + synthetic + street view data. The recognition results of different methods with different combinations of data are shown in Fig 3.8. The ACF method does not work well when only the given training data is used since the data is not enough for the 7 different models. The method [3] works well when only training data is available, but the result does not increase as significantly as other methods when more data is available. On the contrary, when the single-task CNN is used in [3] with only the given training data, it does not work well, but it will be much better than the shallow

CNN in [3] when more data is used. The main reason is that structure of shallow CNN in [3] is not deep enough to learn from more data.

In conclusion, both the synthetic signs and street view images can increase the recognition score of all methods. Street view images are more useful than synthetic signs, and the combination of the two types of data gets the best result on all methods.

3.4.6 Effect of video sequence

Recognition results of all video frames are fused to get the final output in our proposed method. In this subsection, we evaluate the effect of fusion. For this purpose, we first show the results when only one frame is used. For comparison, we also show the score when fusing the recognition results from all frames in the rightmost column. We can see that the best score is obtained when all frames are used, which is somehow obvious. One interesting thing we observe is that the score obtained from the frame in the latter part of a video is generally better than the former one. This can be attributed to the fact that the traffic signs in the latter part of a video are more distinct than others.

We have to point out that such a fusion method based on the property of the data set, i.e., all frames in a short video contain the same categories of traffic signs. In the actual usage, although we can acquire a similar short video by taking the current frame and its previous frames together, it is possible that the traffic signs may exist in only a small part of frames in this video. As a result, it is better to modify such a fusion method and pay more attention to the result from the current frame (i.e., the last frame in the short video) because it leads to the best single frame performance.

3.4.7 Implementation detail

The final traffic sign recognition system is written in C++ with OpenCV and Caffe library. OpenCV is used to implement video reading, MSER ROIs extraction and some other basic image processing operations. The trained CNN model is integrated into the system with the support of the C++ interface of Caffe. The input is a short video with 5 to 20 frames. The image size of a frame image is 1280 1024, however, all traffic signs are in the top half of the image, and therefore the running time of the system on frames with half size is also reported. We run our recognition system on a PC with a 4-core 4.0 GHz CPU and a TITAN Black GPU. We can know that most of the time is spent in CNN recognition and ROIs extraction. The cost time of ROIs extraction is influenced by the image size, and more time is needed for larger image. The CNN recognition time is related to the number of ROIs, which is similar for images of full size and half size since the ROIs are almost in areas around the target. Each frame is processed by the system, and results of all frames are fused to get the final result. The average time for a short video with frame size of 1280512 and 12801024 are 5.24s and 6.44s, respectively. The CPU and GPU memory cost of the recognition system are about 750M and 3700M, respectively.

Conclusion

In this paper, we propose a new data-driven system to recognize all categories of traffic signs in low quality short videos captured by a car-mounted camera. The traffic sign ROIs are first extracted using MESRs on multi-channel images. A new multi-task CNN structure is proposed to refine and classify the ROIs in a uniform framework. To train a reliable network, street view data and synthetic images are used to generate larger number of data with low cost. The recognition outputs of all frames are fused to get final result for a video. Our system gets the state-of-the-art result on a challenging new data set.

Since the data plays a vital role in machine learning based method for automatic traffic sign recognition, we plan to use a bootstrap method to automatically label more data with the API of the street view map in the future. In addition, how to detect and recognize traffic signs together with an end-to end framework will be studied in the future work. Finally, increasing the speed of our system and post-processing with geometry information are also considered part of future work.

REFERENCES

[1] 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. (IJCNN), Jul. 2011, pp. 14531460.

[2] 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. (IJCNN), Aug. 2013, pp. 18.

[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. 20222031, Jul. 2016.

[4] H. Gmez-Moreno, S. Maldonado-Bascn, P. Gil-Jimnez, and S. Lafuente-Arroyo, Goal evaluation of segmentation algorithms for traffic sign recognition, IEEE Trans. Intell. Transp. Syst., vol. 11, no. 4, pp. 917930, Dec. 2010.

[5] A. Mgelmose, D. Liu, and M. M. Trivedi, Detection of U.S. traffic signs, IEEE Trans. Intell. Transp. Syst., vol. 16, no. 6, pp. 31163125, Dec. 2015.

[6] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, Traffic sign recognitionHow far are we from the solution? in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Aug. 2013, pp. 18.

[7] Z. Huang, Y. Yu, S. Ye, and H. Liu, Extreme learning machine based traffic sign detection, in Proc. Int. Conf. Multisens. Fusion Inf. Integr. Intell. Syst. (MFI), 2014, pp. 16.

[8] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, A robust, coarseto- fine traffic sign detection method, in Proc. Int. Joint Conf. Neural Netw. (IJCNN), 2013, pp. 15.

[9] Y. Zhu, X. Wang, C. Yao, and X. Bai, Traffic sign classification using two-layer image representation, in Proc. 20th IEEE Int. Conf. Image Process. (ICIP), Sep. 2013, pp. 37553759.

[10] M. Jaderberg et al., Spatial transformer networks, in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 20082016.