CTSD: A Dataset for Traffic Sign Recognition in Complex Real-World Images

Yanting Zhang, Ziheng Wang, Yonggang Qi, Jun Liu, Jie Yang

Beijing University of Posts and Telecommunications(BUPT), Beijing, China, 100876

***Abstract*—Traffic sign recognition(TSR) is an indispensable component for vision-based system of self-driving car. Promising results have been achieved which especially benefit from the rapid development of deep neural networks recently. However, there are few works focusing on the algorithms' performances towards different complex conditions, such as weather and viewpoint variations. In this paper, we propose a new real-world TSR dataset, which is a dataset with several fine-grained conditions fine labeled involving weather, light condition, occlusion, distance, color fading and camera angle. Detailed and unbiased comparison results are reported about the performances of several state-of-the-arts on our proposed and five public TSR datasets. Experimental results demonstrate that current arts for TSR are still far from satisfactory especially when it comes to complex real-world cases.**

***Index Terms*—Traffic sign recognition(TSR), traffic sign dataset(TSD)**

I. Introduction

Recently, a prosperity has been observed in the development of automatic driving and driver assistance systems. TSR acts as a critical role in the integration of car vision system, providing drivers with safe and efficient navigation. TSR dataset plays an critical role in developing TSR algorithms at both training and testing stages. Therefore, multiple public traffic sign datasets(TSDs) have been collected in the community over the years, such as UAH Dataset[6], BelgiumTS Dataset(BTSD)[2], CVL Dataset[7], German Traffic Sign Detection Benchmark(GTSDB)[1], LISA[3], Russian Traffic Sign Dataset(RTSD)[4], and Tsinghua-Tencent 100K(TT100K)[5].

Although traffic sign datasets like GTSDB, BTSD, LISA, RTSD and TT100K have detailed annotations about bounding boxes and sign types, they don't explicitly classify and annotate the image conditions like weather and camera angle. They are not specially designed for evaluating TSR algorithms in fine-grained various conditions. UAH classifies images into different conditions like occlusion, rotation, and color. But it doesn't provide such annotations, which makes it difficult to evaluate on some special cases, such as the performance in raining. Thus, a more complex real-world dataset is needed to evaluate the performance of different algorithms. It should be well organized and take the complexity of real-world conditions and diversity of traffic signs into consideration.

The problem of TSR is usually formulated into two stages: detection and classification, where poor quality of traffic sign images, within-class variability and between-class similarity always pose challenges. Traditional methods are relying on hand crafted features like Scale-invariant Feature Transform(SIFT), Histogram of Oriented Gradient(HOG)[8], and Local Binary Patterns(LBP), which can be combined with machine learning methods such as Support Vector Machine(SVM) and boosting for classification. The modern history of object recognition goes along with the development of Convolutional Neural Networks(ConvNets). Deep-learning-based methods gradually become mainstream. Researchers propose kinds of methods for object recognition, such as Region-based Convolutional Neural Networks(R-CNN)[9], OverFeat[10], SPPNet, MultiBox, Faster R-CNN[11], SSD[12], YOLOv2[13], and so on. These methods shed light on recognizing traffic signs by using deep-learning-based approaches[14].

All in all, pinpointing a traffic sign's location and its type in real-world images is a challenging computer vision task[5]. It is of high industrial relevance in constructing advanced driver assistance systems. Although plenty of algorithms have been proposed, there is no clear consensus on which one is better in dealing with various complex real-world images. This can be accounted for a lack of comprehensive comparisons among these methods regarding to recognizing traffic signs in different external conditions. In this paper, we provide experimental comparisons of several fundamental methodologies used for TSR on our proposed and five public TSDs. The main contributions can be concluded as follows:

(i) We introduce a new real-world dataset for TSR, called CTSD. Images cover large variations in weather, illuminance, occlusion, distance, color fading/stains, and viewpoint. To our best knowledge, no such dataset has been proposed before with so many different conditions fine classified and labeled.

(ii) We provide a survey on publicly available datasets for TSR and give a comparison about their scales, categories, and limitations. In particular, these datasets are used to train and test kinds of models, hence we try to unlock the value of the data and discover how they suit the real-world road situation.

(iii) We investigate into plenty of technologies regarding to object detection. Both traditional methods and deep-learning-based methods are benchmarked.

II. Dataset

We begin by giving a survey on some popular TSDs. And a new dataset CTSD is proposed to better evaluate numerous advanced techniques for TSR in different conditions.

978-1-5386-4458-4/18/$31.00 ©2018 IEEE

TABLE I
THE OVERVIEW OF FIVE PUBLIC DATASETS(W-WEATHER, L-LIGHT, O-OCCLUSION, D-DISTANCE, C-COLOR FADING, V-VIEWPOINT).

Name	Region	Year	Scale	Types	Image Size	Sign Size	Conditions annotated? W	L	O	D	C	V
GTSDB[1]	German	2013	900	43	1360×800	16×16 ∼ 128×128						
BTSD[2]	Belgium	2010	9006	62	1628×1236	5×11 ∼ 983×653						
LISA[3]	America	2012	6610	47	640×480 ∼ 1024×522	6×6 ∼ 167×168						
RTSD[4]	Russia	2016	104359	156	1280×720 ∼ 1920×1280	16×16 ∼ 307×288						
TT100K[5]	China	2016	10000	182	2048×2048	6×6 ∼ 438×492						
Our Dataset	China	2018	2205	153	600×600 ∼ 2048×2048	8×10 ∼ 1186×1000	√	√	√	√	√	√

Fig. 1. Traffic sign patterns in China

Fig. 2. Traffic sign samples with a 6-digit number depicting different conditions whose meanings are shown in "Labeling Rule" table. For example, the number of "210100" means the conditions of cloudy weather, good light, no occlusion, close to camera, no color fading, and oblique view angle.

A. Public Datasets for TSR

Table I gives the overview of public traffic sign datasets. **RTSD**[4] is the largest dataset. **BTSD**[2] is produced by roof-mounted cameras, which also provides a chance for multi-view traffic sign recognition. **LISA**[3] contains annotated frames as well as original videos taken by various camera types. These TSDs are created based on several video sequences and driving scenarios seldom change. High similarity exists among images in spite of quite a few pictures. **GTSDB**[1] addresses above issues and serves as a benchmark for single-image German traffic sign detection problem. Comprising only 900 images selected from sequences recorded on several tours, it still captures urban, rural and highway scenarios in daytime/dusk and various weather conditions. **TT100K**[5] collects images from Tencent Street Views. It covers downtowns and suburbs from 5 cities in China. TT100K annotates traffic signs' bounding boxes, classes, and pixel masks. However, Signs are all pretty small in high resolution pictures.

B. Our Proposed Dataset for TSR

Data Collection: Images in our dataset come from three sources: 1) Videos from online websites like *Youku*. 2) Driving recorders. 3) Phone's cameras. Several volunteers in our lab have taken videos along roads using phone's camera. We then extract frames from collected video sequences by hand, with full consideration of reducing redundancy and ensuring the quality of selected images. Relevant traffic signs visible in the images are labeled manually. Finally, there are totally 2205 annotated pictures with 3755 traffic signs.

Sign classes: The majority of Chinese traffic signs can be divided into three categories according to usages: "warning" (yellow triangle with black boundary), "mandatory" (circles, blue ground with white content), and "prohibition" (white ground with red circular border), as shown in Fig.1. They are further divided into subclasses with similar geometric shape and appearance but different details. There are 153 subclasses of traffic signs in CTSD and we adopt the same naming rule of signs' subclasses with [5]. Imbalance among classes also exists in our dataset. Unusual signs possess a small proportion.

Sign conditions: Our dataset covers large variations in kinds of complex real-world conditions. Fig. 2 shows some examples. Different conditions of images are marked by a 6-digit number, representing the status of "weather, light, occlusion, distance, color fading, viewpoint". They all bring about challenges in recognizing process. Although current existing public TSDs contain some different driving scenarios, they haven't labeled the images based on distinct conditions, as shown in Table.I. It's inconvenient to use them to test models' performances under targeted conditions. A dataset with kinds of realistic conditions like ours, will hopefully provide an opportunity for researches to do TSR in different complex real-world environments.

III. BASELINE METHODS

Haar+Adaboost: In [15], Viola and Jones proposed a cascade structure for detection utilizing a simple and efficient classifier of AdaBoost for automatic feature selection based on Haar-like features. The approach is firstly used for real-time face detection, but can be easily transferred to TSR.

HOG+SVM: Substantial gains have been obtained with the adoption of gradient-based features. In [8], Dalal and Triggs popularize HOG features for human detection and employ a classifier of SVM.

OverFeat-based: By using a regular pattern of "region proposal", OverFeat can reuse convolution computations from each layer, requiring only a single forward pass for inference. The detector has shown how a multi-scale and sliding window

TABLE II
RESULTS OF DIFFERENT BASELINE METHODS ON VARIOUS DATASETS(P-PRECISION, R-RECALL).

Methods/Datasets	GTSDB		BTSD		LISA		RTSD		TT100K		**Our Dataset**	
	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)
Haar+AdaBoost[15]	17.9	13.9	33.0	21.1	38.2	11.9	37.1	21.6	7.8	22.7	15.3	20.8
HOG+SVM[8]	75	13.9	59.3	21.1	63.4	11.9	1.3	21.6	74.9	22.7	51.2	20.8
OverFeat-based[5]	84.9	55.4	65.1	68.6	92.2	87.0	76.5	17.7	77.2	23.1	52.3	58.0
Faster R-CNN[11]	81.7	72.8	90.3	78.1	70.2	25.6	96.3	53.7	63.4	20.6	53.2	31.2
YOLOv2[13]	98.8	39.6	92.7	65.8	99.0	57.7	98.9	12.0	67.5	27.7	60.4	33.6
SSD[12]	100	19.0	76.6	47.1	92.2	31.9	93.1	10.6	75.5	5.2	46.7	21.5

TABLE III
RESULTS OF DEEP-LEARNING-BASED METHODS ON DIFFERENT CONDITIONS(P-PRECISION(%), R-RECALL(%)).

Conditions	Weather										Light				Occlusion				Distance				Color-fading				Facing-camera			
	Sunny		Cloudy		Rainy		Snowy		Foggy		Good		Bad		Yes		No		Far		Close		Yes		No		Yes		No	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
OverFeat-based[5]	57	63	50	59	45	44	56	45	57	54	58	61	48	55	50	54	56	60	58	58	61	53	47	43	53	61	56	53	55	62
Faster R-CNN[11]	54	31	52	31	38	28	82	64	70	43	57	34	45	25	53	24	54	33	50	26	56	37	38	22	55	33	52	31	62	36
YOLOv2[13]	65	35	59	34	44	30	47	52	65	41	61	35	55	32	60	35	67	36	65	37	57	33	55	29	61	35	62	34	55	36
SSD[12]	48	23	44	17	37	26	59	45	60	39	48	26	39	13	34	24	48	22	63	10	44	32	13	5	48	23	46	23	54	18

approach can be efficiently implemented within a ConvNet. Zhu et. has modified this project in detection case[5].

Faster R-CNN: To overcome the bottleneck of region proposal in Fast R-CNN, Ren et al. propose Faster R-CNN. Region proposal networks which computes proposals using a deep net make the object proposal generator share full-image convolutional features with the detection network, allowing the detection system to achieve a fast frame rate.

YOLOv2: The YOLO design unifies separate components into a single neural network, using features from entire image to predict bounding boxes and classes simultaneously. It enables end-to-end training and realtime speeds. YOLOv2 takes additional measures like batch normalization and high resolution classifier to improve performance.

SSD: The key feature of SSD is regarding to the use of multi-scale convolutional bounding box outputs, which are attached to multiple feature maps at the top of the network. SSD completely eliminates proposal generation and subsequent pixel or feature resampling stages. It encapsulates all computation in a single network.

IV. EXPERIMENT

A. Setup

Traditional methods are implemented by Python and Scikit Learn package. To mimic the region proposal step, Selective Search(SS) is used to generate candidate object locations which are then resized to 50×50 pixels. The Haar-like feature descriptors come into "type-4" features in a 10×10 pixel window. We set 1000 Decision Trees with depth 2 as the weak classifiers in the Adaboost algorithm. As for constructing HOG descriptor, we apply 24×24 pixel blocks of nine 8×8 pixel cells and define the orientations of 8. Then a non-linear SVM classifier with "RBF" kernel is applied.

In the deep-learning-based methods, we use stochastic gradient descent(SGD) with momentum of 0.9 and weight decay of 0.0005. We tune ZF net in Faster R-CNN and apply SSD300 model for SSD. To give an objective comparison without prejudice and observe the algorithms' original performances in recognizing traffic signs, we haven't spent much time on adjusting parameters. Besides, for most dataset, we separate it into training and testing sets with about 9:1 ratio to give the algorithms abundant training samples(or 6:3:1 ratio for training:validation:testing set). The experiments are done on a NVIDIA GTX 1080Ti 12GB GPU and the NVIDIA Deep Learning GPU Training System using the Caffe deep learning framework by Berkeley Learning and Vision Center.

B. Results and Analysis

We use precision(P) and recall(R) to evaluate the results. The Intersection over Union (IoU) threshold is set as 0.5 for all cases. In deep-learning-based methods, we compute $P = N_{tp}/N_{rt}$ and $R = N_{tp}/N_{gt}$, where N_{tp} is the number of traffic signs whose location and class are both accurately predicted. N_{rt} and N_{gt} represents the number of traffic signs for predicted result and ground truth, separately. In traditional methods, we define $P = N_{tp}/N_{rt}$ and $R = N_{tb}/N_{gt}$, where N_{tb} represents the number of boxes localized in right position(IoU$\geq$0.5) obtained through SS method.

Table. II shows the results of different methods on various datasets, which could be concluded three-fold: 1) Traditional method of SVM classifier relying on HOG descriptors greatly outperforms the combination of Haar-like feature and Adaboost for most datasets. Because HOG calculates the gradients and fully utilizes available image information revealed by abrupt edges while Haar-like feature blurs this. The precision of "HOG+SVM" even has 9 times and 4 times increases than "Haar+Adaboost" for TT100K and our dataset. Adaboost performs bad in imbalanced dataset(TT100K), while SVM is not good at dealing with large-scale dataset(RTSD) and the training consumes too much time. SS couldn't find enough positive proposals which leads to low recall values about 10%$\sim$20%. 2) Deep-learning-based methods perform

Fig. 3. Examples of recognition result of traffic signs. Green boxes indicate exactly the right localization and classification. Red boxes indicate wrong detection. Orange boxes represent traffic signs which haven't been successfully detected in the images.

much better than traditional methods, especially in large-scale datasets like RTSD and BTSD which provide abundant training samples. The development of ConvNets has brought about large gains in object detection. However, for small objects such as traffic signs in TT100K, "HOG+SVM" also have a comparable performance with them. 3) In deep-learning-based methods, Faster R-CNN and Yolov2 achieve higher precisions on large-scale datasets like RTSD and BTSD. Without a follow-up feature resampling step, the classification for small objects is hard for SSD, which has low recall values. YOLOv2 has applied some measures such as batch normalization, high resolution classifier, and convolutional with anchor boxes, which makes its recall values comparable to those region-proposal-based methods like Faster R-CNN and OverFeat.

From dataset perspective, it has been observed that the more complex the images are, the more difficult it is to recognize traffic signs inside. For example, although TT100K has a great number of pictures, traffic signs are pretty small in high-resolution pictures and the distribution of sign classes is severely imbalanced, which leads to a lower precision and recall value compared with the performance of same methods on other datasets. Moreover, the images in our dataset has been labeled with different conditions which provides an opportunity for us to observe the performances of algorithms in various complex real-world conditions. We show how deep-learning-based methods perform regarding to different conditions in Table. III, and Fig.3 gives some examples of recognition result. Signs in small size or with color fading/occlusions are challenging for detection. The camera angle and bad weather like rainy days also bring about difficulty in recognizing.

V. Conclusion

In this paper, we introduce a new real-world dataset for TSR. Compared with other public TSDs, the traffic signs in our dataset cover various conditions of weather, light, occlusion, color fading, distance, viewpoint, and etc. The experimental results on our proposed dataset illustrated that recognizing traffic signs is still very challenging especially in complex real-world conditions like bad weather or oblique viewpoint. And the proposed dataset provide a benchmark for special cases. However, the scale is not big yet. In future work, we will collect more data to enrich our dataset.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (61601042, 61671078, 61701031), 111 Project of China (B08004, B17007), and Center for Data Science of Beijing University of Posts and Telecommunications. We gratefully acknowledge the support of NVIDIA Corporation for the donation of the GPUs used for this research.

References

[1] S. Houben and et al., "Detection of traffic signs in real-world images: The german traffic sign detection benchmark," in *IJCNN*. IEEE, 2013, pp. 1–8.

[2] R. Timofte and et al., "Multi-view traffic sign detection, recognition, and 3d localisation," *Machine vision and applications*, vol. 25, no. 3, pp. 633–647, 2014.

[3] A. Mogelmose and et al., "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.

[4] V. I. Shakhuro and et al., "Russian traffic sign images dataset," *Computer Optics*, vol. 40, no. 2, pp. 294–300, 2016.

[5] Z. Zhu and et al., "Traffic-sign detection and classification in the wild," in *CVPR*, 2016, pp. 2110–2118.

[6] S. Maldonado-Bascón and et al., "Road-sign detection and recognition based on support vector machines," *IEEE transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 264–278, 2007.

[7] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition," in *Scandinavian Conference on Image Analysis*. Springer, 2011, pp. 238–249.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.

[9] R. Girshick and et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014, pp. 580–587.

[10] P. Sermanet and et al., "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[11] S. Ren and et al., "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[12] W. Liu and et al., "Ssd: Single shot multibox detector," in *ECCV*. Springer, 2016, pp. 21–37.

[13] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[14] Á. Arcos-García and et al., "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, pp. 158–165, 2018.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, vol. 1. IEEE, 2001, pp. I–I.