Traffic-Sign Detection and Classification in the Wild

Zhe Zhu
TNList, Tsinghua University
Beijing, China
ajex1988@gmail.com

Dun Liang
TNList, Tsinghua University
Beijing, China
randonlang@gmail.com

Songhai Zhang
TNList, Tsinghua University
Beijing, China
shz@tsinghua.edu.cn

Xiaolei Huang
Lehigh University
Bethlehem, PA, USA
huang@cse.lehigh.edu

Baoli Li
Tencent
Beijing, China
barneyli@tencent.com

Shimin Hu
TNList, Tsinghua University
Beijing, China
shimin@tsinghua.edu.cn

Abstract

Although promising results have been achieved in the areas of traffic-sign detection and classification, few works have provided simultaneous solutions to these two tasks for realistic real world images. We make two contributions to this problem. Firstly, we have created a large traffic-sign benchmark from 100000 Tencent Street View panoramas, going beyond previous benchmarks. It provides 100000 images containing 30000 traffic-sign instances. These images cover large variations in illuminance and weather conditions. Each traffic-sign in the benchmark is annotated with a class label, its bounding box and pixel mask. We call this benchmark Tsinghua-Tencent 100K. Secondly, we demonstrate how a robust end-to-end convolutional neural network (CNN) can simultaneously detect and classify traffic-signs. Most previous CNN image processing solutions target objects that occupy a large proportion of an image, and such networks do not work well for target objects occupying only a small fraction of an image like the traffic-signs here. Experimental results show the robustness of our network and its superiority to alternatives. The benchmark, source code and the CNN model introduced in this paper is publicly available[1].

1. Introduction

Scene understanding is the ultimate goal of computer vision; detecting and classifying objects of various sizes in the scene is an important sub-task. Recently, deep learning methods have shown superior performance for many tasks such as image classification and speech recognition. One particular variant of deep neural networks, convolutional neural networks (CNNs), have shown their strengths for tasks including image classification, localization and detection. Two benchmarks widely used to evaluate detection performance are PASCAL VOC [7] and ImageNet ILSVRC [20]. In these datasets, target objects typically occupy a large proportion of each image (the bounding box of each object of interest fills on average about 20% of the image). However, for some tasks, objects of interest may only occupy a small fraction of an image, such as traffic-signs in images captured while driving. A typical traffic-sign might be say 80×80 pixels, in a 2000×2000 pixel image, or just 0.2% of the image. In fact, many tasks require detection and classification of small but significant objects, so it is important to devise and evaluate methods which perform well when the object of interest is not the main, or even a major, scene item.

Traffic signs may be divided into different categories according to function, and in each category they may be further divided into subclasses with similar generic shape and appearance but different details. This suggests traffic-sign recognition should be carried out as a two-phase task: detection followed by classification. The detection step uses shared information to suggest bounding boxes that may contain traffic-signs in a specific category, while the classification step uses differences to determine which specific kind of sign is present (if any). (We note that the words 'detection' and 'classification' have different meanings in the general object recognition community where, as exemplified by the ImageNet competition, classification means giving an *image* a label rather than an object, and detection means finding the bounding box of an object in a specific category.)

Since the launch of the German traffic-sign detection and classification benchmark data[24, 25], various research groups have made progress in both the detection bench-

[1] http://cg.cs.tsinghua.edu.cn/traffic-sign/

mark(GTSDB) [25] task and classification benchmark (GTSRB) [24] task. Current methods achieve perfect or near-perfect results for both tasks, with 100% recall and precision for detection and 99.67% precision for classification. While it may appear that these are thus solved problems, unfortunately, this benchmark data is not representative of that encountered in real tasks. In the GTSDB detection benchmark task, the algorithms must only detect traffic-signs in one of 4 major categories. In the GTSRB classification benchmark, the traffic-sign occupies most of the image, and the algorithms must only decide which subclass the sign belongs to; furthermore; there are no negative samples disrupting the classification. In real world tasks, the main difficulty when detecting and classifying traffic-signs in an ordinary image is their very small size, often less than 1% of the image. The potential candidate regions are orders of magnitude smaller than in PASCAL VOC and ImageNet ILSVRC. Furthermore, the algorithm must filter out many potential negative cases while retaining true traffic-signs. We have thus created a new, more realistic benchmark, and have also used it to evaluate a combined CNN approach to traffic sign detection and classification.

The contributions of this paper are as follows.

- We have created a new, more realistic traffic-sign benchmark. Compared to the widely used detection benchmark GTSDB, our benchmark contains 111 times as many images, at 32 times the image resolution. The traffic-signs in our benchmark cover real-world conditions, with large variations in such aspects as illuminance and weather conditions, also including examples with occlusion. Our benchmark is, unlike previous ones, annotated with a pixel mask for each traffic-sign, as well as giving its bounding box and class. We call this benchmark Tsinghua-Tencent 100K.

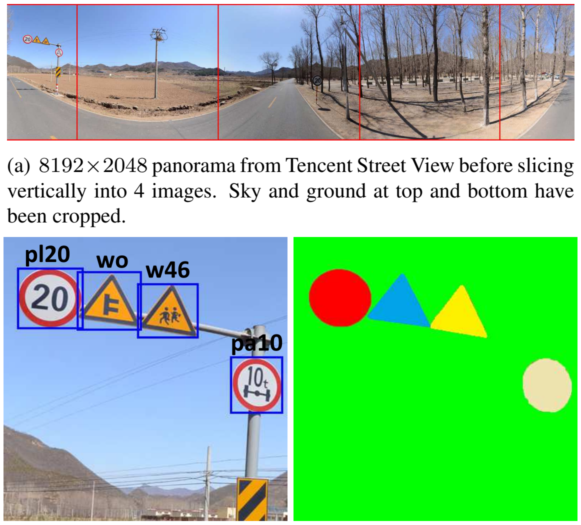
- We have trained two CNNs for detecting traffic signs, and simultaneously detecting and classifying traffic-signs. Evaluation on our benchmark shows the robustness of the two networks.

The rest of the paper is organized as follows: in Section 2 we discuss related work. Details of our benchmark are given in Section 3, while the architecture of our network is presented in Section 4. We give experimental results in Section 5 and conclusions in Section 6.

2. Related work

2.1. Traffic Sign Classification

Before the widespread adoption of convolutional neural networks, various object detection methods were adapted for traffic-sign classification, e.g. based on SVMs [18] and sparse representations [17]. Recently, convolutional neural network approaches have been shown to outperform such simple classifiers when tested on the GTSRB benchmark. These approaches include using a committee of CNNs [4], multi-scale CNNs [22] and CNNs with a hinge loss function [14], the latter achieving a precision rate of 99.65%, better than human performance [25]. However, as noted earlier, these approaches perform classification on already detected signs, which is impractical in real applications.



(a) 8192×2048 panorama from Tencent Street View before slicing vertically into 4 images. Sky and ground at top and bottom have been cropped.

(b) Bounding box and class label (c) Pixel mask annotation

Figure 1. Our benchmark contains 100000 high resolution images in which all traffic-signs are annotated with class label, bounding box, and pixel mask. The images are cut from from Tencent Street Views which contain realistic views traffic-signs in their environments.

2.2. Object Detection by CNNs

After interest in CNNs was initially rekindled by their use in [15] for image classification, they were quickly adapted to object detection. In *OverFeat* [21], Sermanet et al. observed that convolutional networks are inherently efficient when used in a sliding window fashion, as many computations can be reused in overlapping regions. They demonstrated a network that can determine an object's bounding box together with its class label.

Another widely used strategy for object detection using CNNs is to first calculate some generic object proposals and perform classification only on these candidates. R-CNN [8] was the first to use this strategy, but it is very slow for two reasons. Firstly, generating category-independent object proposals is costly. *Selective search* [29] takes about 3 s to generate 1000 proposals for the Pascal VOC 2007 images; the more efficient *EdgeBoxes* approach [30] still takes about 0.3 s. Secondly, it applies a deep convolutional network to every candidate proposal, which is very inefficient.

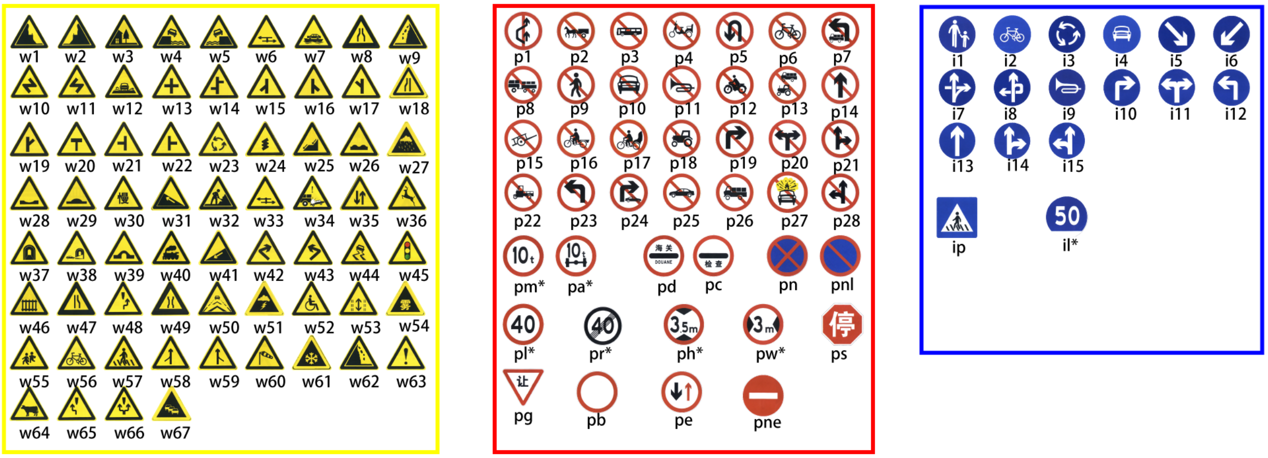


Figure 2. Chinese traffic-sign classes. Signs in yellow, red and blue boxes are warning, prohibitory and mandatory signs respectively. Each traffic-sign has a unique label. Some signs shown are representative of a family (e.g. speed limit signs for different speeds). Such signs are generically denoted above (e.g. 'pl*'); the unique label is determined by replacing '*' by a specific value (e.g. 'pl40' for a 40 kmh speed limit sign).

To improve efficiency, the spatial pyramid pooling network (SPP-Net) [10] calculates a convolutional feature map for the entire image and extracts feature vectors from the shared feature map for each proposal. This speeds up the R-CNN approach about 100 times.

Girshick et al. later proposed Fast R-CNN [9], which uses a *softmax* layer above the network instead of the SVM classifier used in R-CNN. Ignoring object proposal time, it takes 0.3 s for Fast R-CNN to process each image. To overcome the bottleneck in the object proposal step, in Faster R-CNN [19], Ren et al. proposed *region proposal networks* (RPNs) which use convolutional feature maps to generate object proposals. This allows the object proposal generator to share full-image convolutional features with the detection network, allowing their detection system to achieve a frame rate of 5 fps on a powerful GPU.

While these works determine object proposals by hand, Szegedy et al. [27] improved upon a data-driven proposal generation method [6], as well as improving the network architecture, to achieve a frame rate of 50 fps in testing, with competitive detection performance.

However, the performance of all of these object detection networks was evaluated on PASCAL VOC and ILSVRC, where target objects occupy a large proportion of the image.

3. Benchmark

We now explain our new benchmark: where we obtained the data, how we annotated it, and what it finally contains.

3.1. Data Collection

While general image datasets such as ImageNet[5] and Microsoft COCO[16] have been generated by downloading Internet images retrieved by search engines using keywords, relatively few Internet users upload real-world images containing traffic-signs as might be seen in the street, and even when they do, the traffic signs are incidental: such images will not be tagged with the names of any signs they contain. Such an approach cannot be used here. Furthermore, to mimic a real world application scenario, images without traffic-signs should be also included in the benchmark, to evaluate if a detector can distinguish real traffic-signs from other similar looking objects. We determined that an ideal way to collect useful images would be to extract data from Tencent Street Views.

Presently, Tencent Street Views cover about 300 Chinese cities and the road networks linking them. The original panoramas were captured by 6 SLR cameras and then stitched together. Image processing techniques such as exposure adjustment were also used. Images were captured both from vehicles and shoulder-mounted equipment, at intervals of about 10 m. The nature of the images provide two benefits for our benchmark. Firstly, traffic-signs in successive shots are related by a homography. Unlike in GTSRB [25], whose traffic-signs were extracted from a video sequence, leading to many very similar images, the appearances of an instance of a traffic-sign in our benchmark vary significantly. Secondly, an instance of a traffic-sign in successive images helps the participants constructing the benchmark to correctly determine its classes: partially occluded or blurred traffic-signs can be recognized from their

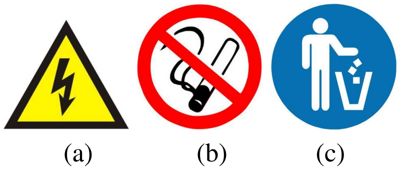


Figure 3. Signs like traffic-signs, but with other meanings.

occurrences in previous or subsequent shots.

To create the benchmark images, the top 25% and bottom 25% of each panorama image was cropped off (as unlikely to contain any signs), and the remainder sliced vertically into 4 sub-images. See Figure 1.

We chose 10 regions from 5 different cities in China (including both downtown regions and suburbs for each city) and downloaded 100000 panoramas from the Tencent Data Center.

3.2. Data Annotation

The images collected were next annotated by hand. Traffic signs in China follow international patterns, and can be classified into three categories: warnings (mostly yellow triangles with a black boundary and information), prohibitions (mostly white surrounded by a red circle and also possibly having a diagonal bar), and mandatory (mostly blue circles with white information). Other signs exist that resemble traffic-signs but are in fact not; some are illustrated in Figure 3. Such signs are placed in an 'other' class of a particular category. During traffic-sign annotation, we recorded the bounding box, boundary vertices and class label for the sign. To determine the pixel mask for the sign, we use two modes: polygon mode and ellipse mode. In polygon mode, we mark the vertices of the polygon while in ellipse mode we mark arbitrary 'vertices' along the boundary of the ellipse, and we fit the shape automatically using the marked vertices. For a triangle sign we only mark three vertices; for distorted signs we may mark additional vertices for accurate segmentation. Circle signs appear as ellipses, unless occluded, so we mark 5 vertices to which we can fit a ellipse during post-processing. The most complicated cases concern occluded signs. In this case, we mark the bounding box, the polygon boundary and ellipse boundary (if appropriate), and intersect them to find the final mask. We illustrate our annotation pipeline in Figure 4, and show a complicated annotation case in Figure 5.

3.3. Dataset Statistics

Our new benchmark has 100000 cropped images after discarding some of the images only containing background. Of these, 10000 contain 30000 traffic-signs in total. Although our source images cover much of China, an imbalance still exists between different classes of traffic-sign in our benchmark. This is unavoidable: classes such as signs to warn the driver to be cautious on mountain roads appear rarely. Instances per class are given in Figure 6; most instances appear in relatively few classes. The image sizes (in pixels) of the traffic-signs is given in Figure 7; note that small traffic-signs are most common.

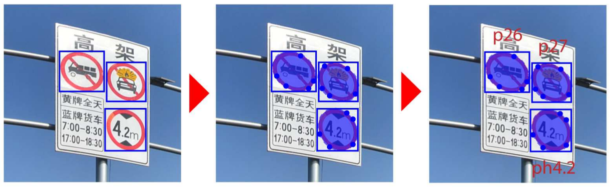


Figure 4. Annotation pipeline. Firstly we locate the traffic-sign and draw its bounding box. Then boundary vertices are marked on the sign's contour to determine the pixel mask. Finally the class label is attached.

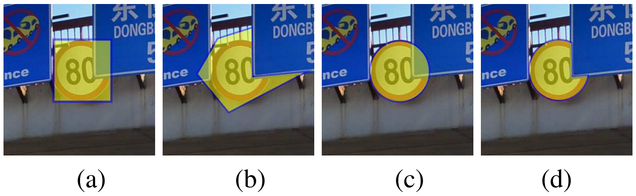


Figure 5. Sign annotation for a complicated case. We mark the bounding box, polygon boundary and circle boundary, and compute their intersection to give the final segmentation mask.

Table 3. Simultaneous detection and classification results for different sizes of traffic signs using Fast R-CNN and our approach. FR: Fast R-CNN recall, FA: Fast R-CNN accuracy, OR: Our method's recall, OA: Our method's accuracy.

Object size	(0,32]	(32,96]	(96,400]
FR	0.24	0.74	0.86
FA	0.45	0.51	0.55
OR	0.87	0.94	0.88
OA	0.82	0.91	0.91

In summary, our newly created benchmark provides detailed annotation for each sign: its bounding box, its pixel mask, and its class. The signs fall into many classes, and there are many instances in many of those classes. The images in this benchmark have resolution 2048×2048. and cover large variations in illuminance and weather conditions. It will hopefully provide a suitable basis for research into both detecting and classifying small objects. We have used it to train our own CNN for this purpose.

4. Neural Network

We trained two networks in total, one for detection alone, and one for simultaneous detection and classification. They share most of the same structure except for the branches in the last layer.

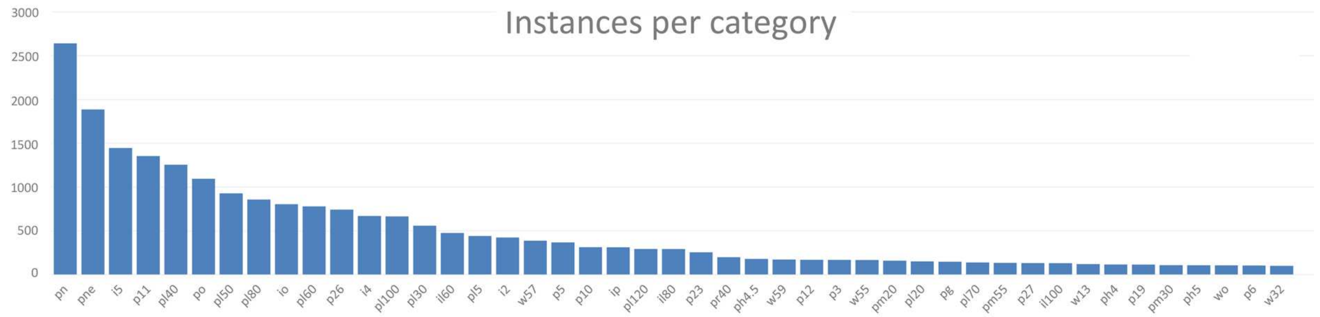


Figure 6. Number of instances in each class, for classes with more than 100 instances.

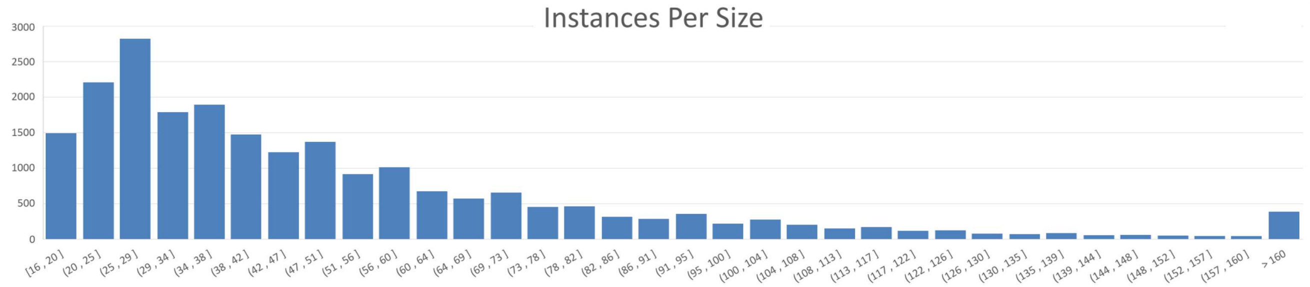


Figure 7. Number of instances of each size.

4.1. Architecture

In [12], Huval et al. evaluate the performance of CNNs on lane and vehicle detection. They use the *OverFeat* framework with a bounding box regression step. Their network is fully convolutional, and the last layer is branched into two streams: a pixel layer and a bounding box layer. Each result output by the pixel layer represents the probability of a certain 4×4 pixel region in the input image containing a target object. For the bounding box layer, each result represents the distance between that region and the four sides of the predicted bounding box of the target. They evaluated their network on a 1.5 hour highway video. Although their network perfectly detected vehicles (regarding all kinds of vehicles as one class), their network can not be directly adapted to train a multi-class detector for small objects, as needed for our problem. Nevertheless, we build upon their network architecture but make noticeable modifications. Firstly, we make the network branch after layer 6, while in [12] branching is done after layer 7. During experiment we found this modification makes the network converge faster compared with precious network structure. As noted in [23, 26], deeper networks perform better: more layers bring more capability. If we let the network branch more earlier, although it has the potential to perform better, this increases training time and consumes more GPU memory, so is not cost-efficient. Thus it is a better balance between speed and accuracy to branch the network after layer 6. Another modification is that our network finally branches into three streams rather than the two streams in [12]. Apart from a bounding box layer and a pixel layer, we added a label layer which can output a classification vector with n elements, where each element is the probability it belongs to a specific class. This allows our network to simultaneously detect and classify traffic signs. Our network architecture is illustrated in Figure 8. More details can be found in our source code. Our implementation uses the Caffe learning framework [13]. When removing the label layer in the 8th layer, this network can be used as a traffic sign detector.

4.2. Training

Due to the uneven numbers of examples of different classes of traffic signs, we used a data augmentation technique during training [14, 22]. We simply ignored classes with fewer than 100 instances. This left 45 classes to classify. Classes with between 100 and 1000 instances in the training set were augmented to give them 1000 instances. Other classes that have more than 1000 instances remain unchanged.

To augment the data, we used the standard template for each class of traffic signs [1], rotated it randomly by an amount in the range $[-20^\circ, 20^\circ]$, scaled it randomly to have size in the range $[20, 200]$, and also added a random but reasonable perspective distortion. We then manually picked images without traffic signs and blended in the transformed template, with additional random noise.

Table 1. Network architecture for multi-class model

layer	data	conv1	conv2	con3	conv4	conv5	conv6	conv7	conv8-bbox	conv8-pixel	conv8-label
output size (chan×h×w)	3,480, 640	96,118, 158	256,59, 79	384,29, 39	384,29, 39	384,29, 39	4096,15, 20	4096,15, 20	256,15, 20	128,15, 20	1000,15 20
input size		3,480, 640	96,59, 79	256,29, 39	384,29, 39	384,29, 39	384,29, 39	4096,15, 20	4096,15, 20	4096,15, 20	4096,15, 20
kernel size, stride, pad		11,4,0	5,1,2	3,1,1	3,1,1	3,1,1	6,1,3	1,1,0	1,1,0	1,1,0	1,1,0
pooling size, stride		3,2	3,2			3,2					
addition		lrn layer	lrn layer				dropout 0.5	dropout 0.5			

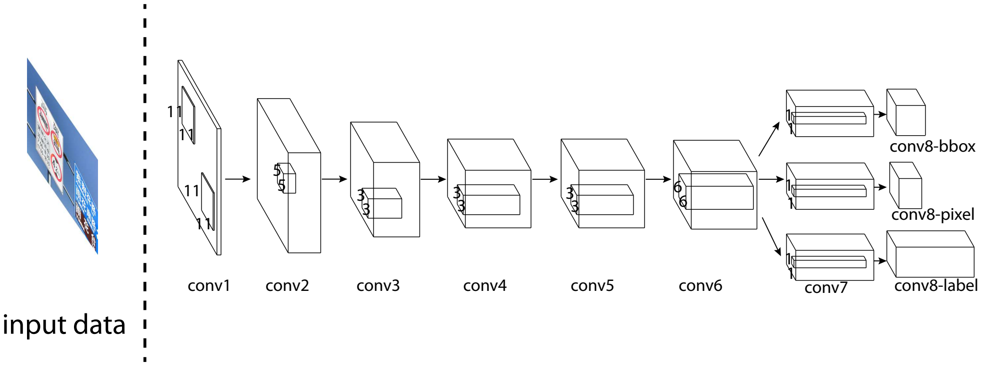


Figure 8. Architecture of our multi-class network. Our network is fully convolutional, and branches after the 6th layer.

5. Results

In our experimental evaluation of our neural network, both training and testing were done on a Linux PC with an Intel Xeon E5-1620 CPU, two NVIDIA Tesla K20 GPUs and 32GB memory. For 10000 panoramas containing traffic-signs, we separated them into a training set and a testing set (as explained in the released benchmark), with about 2:1 ratio to give the deep learning methods plenty of training samples. The other 90000 panoramas were included during testing.

We used the evaluation metrics used for the Microsoft COCO benchmark, and divided the traffic-signs into three categories according to their size: small objects(area $< 32^2$ pixels), medium objects($32^2 <$ area $< 96^2$) and large objects (area $> 96^2$). This evaluation scheme can tell the ability of a detector on different sizes of object.

5.1. Detection

We now consider how well various object proposal methods work, as well as our detection network, using our benchmark. A common feature of most popular object detection networks [10, 19, 9] is that they rely on generic object proposals. If the target object is missing from the proposal set, later steps are in vain. The sizes of objects of interest in our benchmark are much smaller than in previous benchmarks, and typical object proposal methods do not work well for such small objects, resulting in traffic-sign candidates of low quality, as we now show. Selective Search [29], Edge Boxes [30] and Multiscale Combinatorial Grouping(MCG) [2] are suggested in [11] to be the most effective object proposal methods. Since MCG is memory intensive and our images are of high resolution, we evaluated the proposal performance for traffic-signs using Selective Search, Edge Boxes and BING [3] instead. Note that Selective Search and Edge Boxes do not need training data, so we directly evaluated their performance on the test set of our benchmark. We trained BING using the same augmented training set as used for our network. The results are illustrated in Figure 9. The average recall for all 10000 proposals for those three approaches is under 0.7. This indicates that object proposal approaches are not suitable for locating small objects in large images, even when the number of candidate proposals is sufficiently large.

Instead, treating all traffic-signs as one category, we trained a detection network using our architecture. Our network achieved 84% accuracy and 94% recall at a Jaccard similarity coefficient of 0.5, without carefully tuning its parameters, which significantly outperforms the results obtained by previous objection detection methods. Also, we note that our network performs in essence just as well for each size of objects.

We also tested our detector on the 90000 panoramas that contained no traffic-signs, and the network perfectly identi-

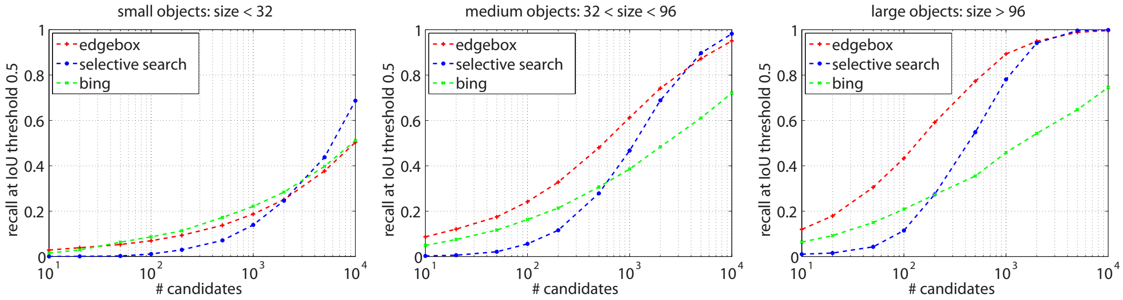


Figure 9. Object proposal results for traffic-signs for various object location methods, for small, medium and large signs.

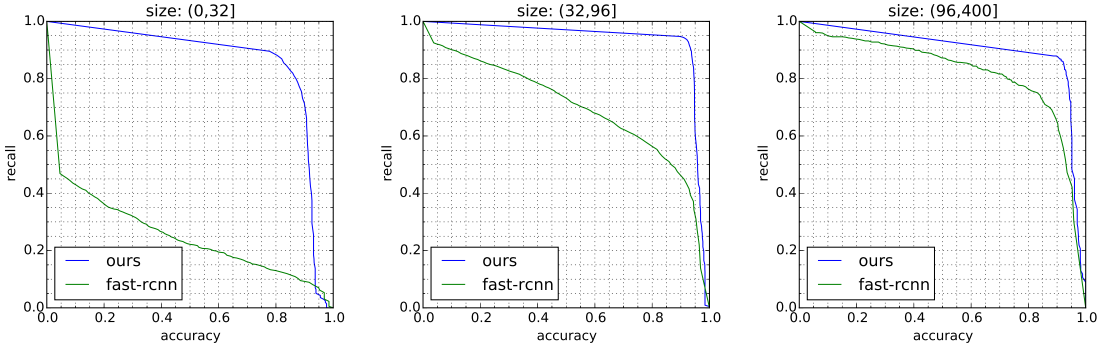


Figure 10. Simultaneous traffic sign detection and classification results Fast R-CNN and our approach, for small, medium and large signs.

fied them all as only containing background.

Further details are provided in the supplemental material.

5.2. Simultaneous detection and classification

We now turn to the combined problem of both finding traffic signs and classifying them. We compare the results provided by our network and the state-of-the-art Fast R-CNN method [9]. (We could not readily make a comparison with Faster R-CNN [19] or the detector in [28] as source code is unavailable). We also generate 10000 proposal for each image when we run the test of Fast R-CNN. The results are given in Figure 10. This clearly indicates that our approach outperforms R-CNN, especially when traffic signs are small. We also give the accuracy and recall for each category for Jaccard similarity coefficient 0.5 in Table 2, and the average accuracy and recall for different object sizes in Table 3; Fast R-CNN has better performance for larger objects. Overall, Fast R-CNN has a recall 0.56 and accuracy 0.50 while our approach has a recall 0.91 and accuracy 0.88.

6. Conclusions

We have created a new benchmark for simultaneously detecting and classifying traffic signs. Compared with previous traffic sign benchmarks, images in this benchmark are more variable, and signs in these images are much smaller. It contains more images than previous benchmarks, and the images have a higher resolution. Furthermore, pixel-wise segmentation of signs is provided. This benchmark provides a new challenge for the traffic sign recognition community. We have trained two networks on this benchmark: one treats all sign classes as a single category and can be regarded as a traffic sign detector. The other network can simultaneously detect and classify traffic signs. Both significantly outperform previous work, and can be used as a baseline for future research. To assist research in this field, we make this benchmark, trained models and source code public available.

In future, we plan to seek out more traffic signs of the classes that rarely appear in this benchmark. We also plan to accelerate the speed of the process in order to run it on mobile devices in real-time.

Acknowledgements

We thank Ralph Martin, Mingming Cheng and the anonymous reviewers for the valuable discussions. We thank Jiaming Lu for his work in the experiment. This work was supported by the Natural Science Foundation of China (Project Number 61120106007,61521002,61373069), Research Grant of Beijing Higher Institution Engineering Research Center, and Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology. Songhai Zhang is the corresponding author.

Table 2. Simultaneous detection and classification results for each class using Fast R-CNN and our approach. FR: Fast R-CNN recall, FA: Fast R-CNN accuracy, OR: Our method's recall, OA: Our method's accuracy.

Class	i2	i4	i5	il100	il60	il80	io	ip	p10	p11	p12	p19	p23	p26	p27
FR	0.32	0.61	0.69	0.41	0.8	0.39	0.65	0.67	0.51	0.44	0.48	0.79	0.70	0.60	0.60
FA	0.68	0.62	0.71	0.52	0.63	0.76	0.51	0.48	0.54	0.69	0.73	0.67	0.94	0.67	0.67
OR	0.82	0.94	0.95	0.97	0.91	0.94	0.89	0.92	0.95	0.91	0.89	0.94	0.94	0.93	0.96
OA	0.72	0.83	0.92	1.00	0.91	0.93	0.76	0.87	0.78	0.89	0.88	0.53	0.87	0.82	0.78

Class	p3	p5	p6	pg	ph4	ph4.5	ph5	pl100	pl120	pl20	pl30	pl40	pl5	pl50	pl60
FR	0.40	0.72	0.54	0.89	0.42	0.83	0.31	0.82	0.57	0.25	0.43	0.48	0.65	0.29	0.42
FA	0.62	0.92	0.66	0.44	0.94	0.75	0.63	0.81	0.91	0.88	0.73	0.85	0.89	0.76	0.86
OR	0.91	0.95	0.87	0.91	0.82	0.88	0.82	0.98	0.98	0.96	0.94	0.96	0.94	0.94	0.93
OA	0.80	0.89	0.87	0.93	0.94	0.88	0.89	0.97	1.00	0.90	0.90	0.89	0.84	0.87	0.93

Class	pl70	pl80	pm20	pm30	pm55	pn	pne	po	pr40	w13	w32	w55	w57	w59	wo
FR	0.23	0.40	0.53	0.63	0.79	0.59	0.77	0.29	0.98	0.32	0.29	0.50	0.56	0.67	0.32
FA	0.67	0.76	0.93	0.77	0.57	0.86	0.21	0.33	0.10	0.36	0.30	0.70	0.38	0.53	0.16
OR	0.93	0.95	0.88	0.91	0.95	0.91	0.93	0.67	0.98	0.65	0.71	0.72	0.79	0.82	0.45
OA	0.95	0.94	0.91	0.81	0.60	0.92	0.93	0.84	0.76	0.65	0.89	0.86	0.95	0.75	0.52

References

[1] Chinese traffic sign template, howpublished = `http://www.bjjtgl.gov.cn/jgj/jgbz/index.html`, note = accessed on 2015-11-7.

[2] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[3] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.

[4] D. C. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *International Joint Conference on Neural Networks*, pages 1918–1921, 2011.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. *CoRR*, abs/1312.2249, 2013.

[7] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015.

[11] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC*, 2014.

[12] B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F. Mujica, A. Coates, and A. Y. Ng. An empirical evaluation of deep learning on highway driving. *CoRR*, abs/1504.01716, 2015.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[14] J. Jin, K. Fu, and C. Zhang. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1991–2000, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[16] T. Lin, M. Maire, S. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[17] K. Lu, Z. Ding, and S. Ge. Sparse-representation-based graph embedding for traffic sign recognition. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1515–1524, 2012.

[18] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras. Road-sign detection and recognition based on support vector machines. *Intelligent Transportation Systems, IEEE Transactions on*, 8(2):264–278, June 2007.

[19] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015.

[21] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[22] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (I-JCNN), The 2011 International Joint Conference on*, pages 2809–2813, July 2011.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[24] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1453–1460. IEEE, 2011.

[25] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[27] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *CoRR*, abs/1412.1441, 2014.

[28] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2553–2561. 2013.

[29] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[30] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. European Conference on Computer Vision, September 2014.