

Efficient Traffic Sign Recognition with Scale-aware CNN

Yuchen Yang¹
 linyeglasses@hotmail.com
 Shuo Liu²
 luke.liu@alumni.ubc.ca
 Wei Ma^{*1}
 mawei@bjut.edu.cn
 Qiuyuan Wang³
 wangqiuyuan@pku.edu.cn
 Zheng Liu²
 zheng.liu@ubc.ca

¹ Beijing University of Technology
 100 Pingleyuan, Chaoyang District
 Beijing, China
² University of British Columbia,
 Okanagan
 3333 University Way, Kelowna
 BC, Canada
³ Peking University
 5 Yiheyuan Road, Haidian District
 Beijing, China

Abstract

The paper presents a Traffic Sign Recognition (TSR) system, which can fast and accurately recognize traffic signs of different sizes in images. The system consists of two well-designed Convolutional Neural Networks (CNNs), one for region proposals of traffic signs and one for classification of each region. In the proposal CNN, a Fully Convolutional Network (FCN) with a dual multi-scale architecture is proposed to achieve scale invariant detection. In training the proposal network, a modified Online Hard Example Mining (OHEM) scheme is adopted to suppress false positives. The classification network fuses multi-scale features as representation and adopts an Inception module for efficiency. We evaluate the proposed TSR system and its components with extensive experiments. Our method obtains 99.88% precision and 96.61% recall on the Swedish Traffic Signs Dataset (STSD), higher than state-of-the-art methods. Besides, our system is faster and more lightweight than state-of-the-art deep learning networks for traffic sign recognition.

1 Introduction

Traffic sign recognition (TSR) is key to Advanced Driver Assistance Systems (ADAS) and Intelligent Transport Systems (ITS). Typically, given a road scene image, TSR automatically localizes and recognizes traffic signs in it, thereby reminding human drivers or helping ADAS make decisions. Many TSR methods have been proposed in recent years [1, 2, 3, 4, 5, 6, 7], and validated on some public traffic sign recognition benchmarks [8, 9, 10].

However, there are still many challenging problems for real-world applications. First, road scenes are extremely complicated, because of varying illumination, color deterioration of traffic signs, and the existence of decorations looking similar to traffic signs. In this situation, it is hard for TSR to obtain a high recall rate while producing few false positives

* Corresponding author.

during detection. Second, unlike lab environments, automobiles have limited memory and computing capacity. Thus, the computational complexity of TSR should be low and its required memory resources should be small. Third, the scales of traffic signs captured by moving vehicles vary in a large range. Traffic signs, especially small ones, are easily missed by current TSR systems.

In this paper, we propose a scale-aware Traffic Sign Recognition framework (scale-aware TSR for short), which deals the above problems well. The overview of the proposed framework is illustrated in Figure 1. It consists of two Convolutional Neural Networks (CNNs), a proposal (class-agnostic detection) network to find possible regions of traffic signs and a classification network for classifying each region proposal. The proposal network adopts a dual multi-scale structure. Given an image, a rough image pyramid is constructed beforehand and fed into the network. Inside of the network, there are two output branches, one from a lower layer for small-scale objects and the other from the last layer for large-scale objects. The results of the dual multi-scale proposal network are aggregated and sent to the classification network. The classification network is designed to be lightweight and capable of fusing multi-scale features.

Our major contributions are as follows: (1) A traffic sign recognition framework named scale-aware TSR is proposed. It beats state-of-the-art methods and achieves 99.88% precision and 96.61% average recall rate on Swedish Traffic Signs Dataset (STSD) [24]; (2) Within the framework, we elaborately design a Fully Convolution Network (FCN) with an efficient dual multi-scale structure for region proposals. The network is experimentally verified to be scale-aware, thereby capable of localizing targets of wide-range scales; (3) Moreover, a lightweight classification sub-network with multi-scale feature fusion structure and Inception module [57] is presented and validated.

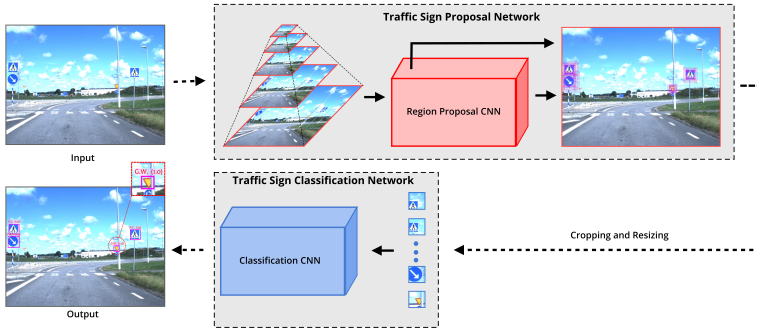


Figure 1: Overview of the proposed scale-aware TSR system.

2 Related Work

In traditional paradigms, TSR is usually carried out as two separate sub-tasks: class-agnostic detection and class-specific classification. They are validated on separate datasets, e.g. the German Traffic Sign Detection Benchmark (GTSDB) [56] for detection and the German Traffic Sign Recognition Benchmark (GTSRB) [55] for classification.

In the task of traffic sign detection, most state-of-the-art methods [24, 56, 40] adopted a general pipeline where a sliding window is firstly employed, followed by a hand-crafted fea-

ture extractor such as HOG [9] or Hough [10], and a classifier (SVM [9] or Random Forests [9]). As the sliding window scheme involves exhaustive search, these methods are too time-consuming to be deployed in real-world scenarios. To tackle this issue, several Regions of Interest (ROIs) based methods [23, 31, 41] were proposed to replace the exhaustive search and gain acceleration. In particular, [41] proposed a color probability model to estimate a probability map from an input image. Based on the probability map, a Maximally Stable Extremal Regions (MSERs) detector is applied to extract traffic sign ROI proposals. This method costs only 67ms per image in GTSDb test dataset. These hand-crafted detectors might not have a promising recall rate when they are applied in new scenes. Most recently, Aghdam *et al.* [4] combined sliding window, feature extractors and classifiers into an end-to-end CNN architecture, which gains improvement on both accuracy and speed. Note that our traffic sign proposal network is partly motivated by this approach.

In the traffic sign classification task, most current methods are based on the assumption that all possible traffic signs have already been detected successfully and their locations are accurate. Before CNNs started to lead state-of-the-art performance, various traditional hand-engineered features along with shallow classifiers were exploited for traffic sign classification, such as pixel-level features + SVM [27] and HOG + Random Forests [44]. Recently, [35] and [36] showed that CNN-based methods achieved better-than-human classification rate. In [2, 8], a committee of CNNs was proposed, where a CNN and a Multi-Layer Perceptron (MLP) are combined together and trained on raw pixel intensities. They obtained a high recognition rate up to 99.15% on GTSRB. Furthermore, [20] proposed a hinge loss stochastic gradient descent method to train CNNs and achieved 99.65% recognition rate on GTSRB. To balance the accuracy and computational efficiency, a small scale CNN was proposed in [44]. It adopted bootstrapping training to enhance its discriminative ability. [15] improved the performance on GTSRB to 99.81% via combination of insights from Inception [37] and the Spatial Transformer Networks [18]. However, these approaches are not lightweight enough for hardware equipment mounted on vehicles. In addition, as mentioned earlier, these methods are tested on already perfectly detected traffic signs. We consider traffic sign class-agnostic detection and classification together for practical applications.

We also researched methods dealing with detection and classification simultaneously, most of which target at generic objects. The most popular framework among them is R-CNN [14]. It uses a pre-trained CNN to extract features from box proposals generated by selective search [38], and then class-specific linear SVMs for classification. The significant advantage of this work is the replacement of hand-engineered features with CNN extracted ones. Meanwhile, some variants of R-CNN were proposed to decrease the running time of R-CNN [13, 17, 30]. It is worth noting that Faster R-CNN [30] amends the multi-stage tasks in R-CNN into an end-to-end detection system which consists of a Region Proposal Network (RPN) as well as a CNN for classification and bounding box regression. Most recently, some TSR works gained insights from generic object detection and developed end-to-end systems. For example, motivated by OverFeat [34], Zhu *et al.* [45] proposed an FCN that can detect and classify traffic signs simultaneously. [42] used FCN [26] and EdgeBoxes [46] to generate region proposals of traffic signs in a coarse-to-fine manner, and then a small size CNN to classify these proposals. This method was evaluated on STSD which is designed for testing the performance on entire TSR systems. We treat this method [42] as our baseline and also evaluate our method on STSD.

3 Methodology

In the following, we describe the two parts in the proposed TSR system (illustrated in Figure 1), region proposal network and classification network, respectively.

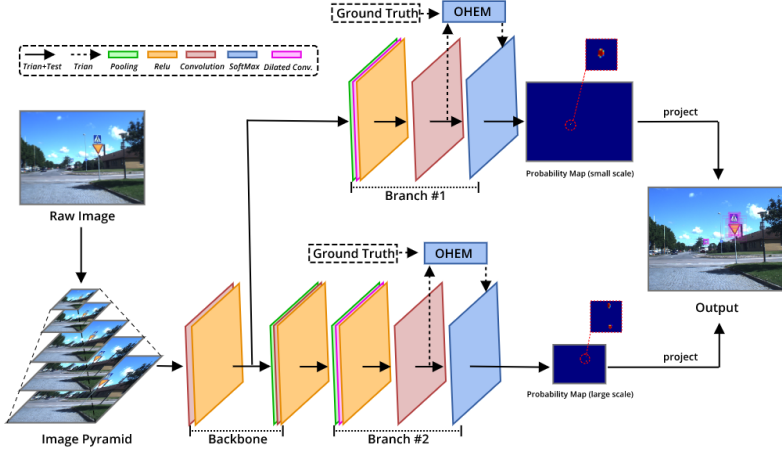


Figure 2: Flowchart and architecture of the proposed DMS-net

3.1 Proposal Network

A Dual Multi-Scale network (DMS-net) is designed for region proposals of traffic signs. Figure 2 illustrates the flowchart and architecture of the DMS-net. Initially, a pyramid with multi-scale images is constructed based on a raw image. The multi-scale pyramid is forwarded through a hierarchical convolutional network layer by layer. For each layer of the image pyramid, the network produces multi-scale probability maps, in which a value represents the probability of its corresponding region in the input image being a part of a traffic sign. The dual multi-scale probability maps are aggregated to determine traffic sign proposals in the form of bounding boxes in the raw image.

Dual Multi-scale Network: Localizing objects in different scales is essential for an outstanding object detector in real-world applications. There are two main strategies to achieve this goal. The first is to train a detector with objects of various scales so that it can localize objects of different sizes in testing images [14]. An alternative approach is to construct a hierarchical structure inside of the detector. Through the structure, the detector draws features of multiple scales to help make decisions. Several state-of-the-art detection methods adopted this strategy, such as MSCNN [9]. In the proposed DMS-net, we combine these two strategies together in the framework of CNN in an efficient way.

As seen in Figure 2, the DMS-net consists of two multi-scale structures, a rough image pyramid (5 scales in our experiments) and a hierarchical network with two output branches. In this way, our DMS-net can detect traffic signs at 10 different scales (5×2) in total with high computation efficiency.

Architecture: The DMS-net adopts an FCN structure which can take an image of arbitrary size as input. This makes our method possible to deal with image pyramids. The network primarily includes a backbone stream and two branches. To be specific, there are

Proposal Network (DMS-net)													
Module Name	Backbone			Branch #1		Branch #2							
Layer	Conv1	Conv2		Conv1_1	Conv1_2	Conv2_1	Conv2_2						
Details	C(60, 9, 9) C(120, 5, 5) st.1, Relu	P(2, 2) C(120, 5, 5) st.1, Relu		P(6, 2)	C(2, 1, 1) st.1	P(4, 2)	C(300, 3, 3) dilation 2 st.1, Relu		C(2, 1, 1) st.1				
				C(300, 2, 2) dilation 3 st.1, Relu									
Classification Network (fusion-net)													
Module Name	Basic Layers		Multi-Scale Features Fusion Module			Inception Module					Classifier Layers		
Layer	Conv1	Conv2	Conv3_1	Conv3_2	Conv3_3	#1*1	#3*3 Reduce	#3*3	#5*5 reduce	#5*5	pool project	Avg Pool	FC
Details	C(100,5,5) st.1, pad 2 Relu	C(150,3,3) st.1, pad 1 Relu	C(250,3,3) st.1, pad 1 Relu	C(250,3,3) st.1, pad 1 Relu	C(250,3,3) st.1, pad 1 Relu	C(64,1,1) st.1, pad 1 Relu	C(32,1,1) st.1, pad 1 Relu	C(32,3,3) st.1, pad 1 Relu	C(16,1,1) st.1, pad 1 Relu	C(32,5,5) st.1, pad 1 Relu	P(3,1) C(96,1,1) st.1, pad 1 Relu	Ave_P(5,1)	FC 11

Table 1: The detailed settings of our DMS-net (top) and fusion-net(bottom). In the details cell, "C (number, size, size)" denotes the number of convolution filters and their local receptive sizes; "st." and "pad" indicate the convolution stride and spatial padding, respectively. "P" indicates average pooling with kernel size and stride.

two convolution layers in the backbone. The first one is followed by a Relu [28] activation function, and the second one is similar except being preceded by a max pooling layer. The two branches have the same structure with a minor difference in configuration. The structure includes a max pooling layer, a dilated convolution layer [47] with the Relu, as well as a 1×1 convolution layer and softmax layer for probability maps. Note that the first branch is out from the first layer of the backbone and the second comes from the end of the backbone. The detailed configurations of our DMS-net are given in Table 1.

Dense Prediction: Another reason to use the FCN structure is its ability to make dense predictions for per-pixel tasks like semantic segmentation [26]. Unlike semantic segmentation [26, 42] where each output pixel is a classifier of its path-connected one in the input image, each value in our probability maps represents the probability of its corresponding receptive field in the input image as parts of traffic signs. In this way, we can generate region proposals with the FCN structure. This idea is mainly motivated by the works in [9, 42].

We take the first branch in Figure 2 and its output as an example to explain the correspondences between probability maps and receptive fields in the input image. Given an image patch of 20×20 pixels, the first convolution layer with kernels of size 9×9 generates 12×12 feature maps. In these feature maps, each value has a 9×9 receptive field in the original image. Then, after the max pooling layer in the first branch with kernels of size 6×6 and stride of 2, the feature maps become 4×4 with a receptive field of size 14×14 . Note that the first convolution layer in the branch is dilated convolution. The dilated convolution has been proved to support exponentially expanding receptive fields without losing resolution or coverage [9, 42]. For more details about dilated convolution, please refer to [47]. In this way, the dilated convolution layer with kernels of size 2×2 and dilation factor of 3 will produce only one value which has a receptive field of the whole image patch (20×20). The last 1×1 convolution layer and softmax producing probability maps will not affect the receptive field. Therefore, for an arbitrary scale image, each value in the probability maps of the first branch corresponds to a region of size 20×20 pixels in the image.

Since there is an additional convolution layer and max pooling layer, the second branch has a larger receptive field. By projecting the multi-scale probability maps into the pyramid images and aggregating them, we obtain dense multi-scale bounding boxes. Note that in our DMS-net, we ignore the bounding boxes with irregular aspect ratios and only keep the square ones. This is because of the observation that most traffic signs stay within square regions.

Training: In order to train the DMS-net, we transform the original ground truth (bounding box style) into spatial probability maps by scanning each image with a square window. If the square window has an intersection-over-union (IoU) with traffic signs higher than 0.7 threshold, we assign 1 (positive) to the objective probability map. If the IoU is lower than

0.3, we assign 0 (negative) to the probability map. Otherwise, we assign a special value, which will be ignored in loss calculation. The size of the scanning window depends on the receptive fields of the specific branches. For each image, we generate 10 ground truth maps with different scales in all. In order to provide a sufficient data to train our network, we adopt several data augmentation tricks including contrast adjustments, random blur, and lighting adjustments. Note that unlike the common patch-level training paradigm in pixel-to-pixel tasks [16, 24], we input the whole images for training efficiently. Because the ratio of negative/positive in the ground truth maps is extremely high, the overall loss will be too small to be backpropagated if we sum over pixel-wise losses in the probability maps. To solve this issue, we adopt the modified Online Hard Example Mining (OHEM) [54] in the training stage to suppress false positives. To be specific, we calculate the pixel-wise loss on all the positions of probability maps. Then we sort the positions by their loss values and select top N positions ($N = 128$ in our experiments). Only the top N selected positions participate in the back-propagation process. This technique guarantees that our DMS-net converges fast and generates few false positives. We train the two branches in our network in a multi-stage way. Specifically, only one branch is trained while the other branch is frozen in each iteration. For training, we use the same standard cross-entropy loss function for each branch with Stochastic Gradient Descent (SGD). The layers are initialized by random weights drawn from a Gaussian distribution with zero mean and 0.01 variance. Other hyper-parameters are momentum 0.9; weight decay 0.0005; batch size 1; initial learning rate 0.001, which is decreased by a factor of 10 after 20k iterations. We trained the model for 50k iterations in our experiments.

3.2 Classification Network

After the region proposals are extracted, we use a classification network, called fusion-net, to differentiate traffic signs of different classes and background. The fusion-net comprises two basic components, a multi-scale feature fusion module and an Inception module [57], followed by an average pooling layer and a fully connected layer. The input image size is 64×64 . The details about our fusion-net are listed in Table 1.

Multi-Scale Features Fusion Module: Fusing features from different layers in a CNN is proved to be effective in improving accuracy in many tasks [2, 21, 26, 52]. Features from different layers have different receptive fields and semantic clues, thereby helping our classification network differentiate fine-grain classes, e.g different speed signs. Our multi-scale feature fusion module has three unified 3×3 convolution layers. The output of each layer is branched out and then concatenated together.

Inception Module and Average Pooling: The Inception module [57] performs multi-size convolution at the same time. All the results are then concatenated. This allows the model to take advantage of multi-level feature extraction from the same input with less time consumption and parameters. For more details about the Inception module, please refer to Table 1 and [57]. After the Inception module, we perform a global average pooling operation as a bridge to connect later with a fully connected layer. This can dramatically help fusion-net reduce overall parameters and prevent over-fitting.

Training: The fusion-net is not trained with DMS-net in an end-to-end manner. Similar to the method in [24], we adopt bootstrapping to mine hard negative examples for training our fusion-net. Instead of sampling training data from the region proposal results, we randomly crop square regions, which have IoU with the traffic signs higher than 0.6 on the raw training images, as positive training data. Regions with the IoU lower than 0.5 are treated as negative

Sign name	FDs[14]		Adaboost+SVR[6]		R-CNN[14]		Faster R-CNN[14]		FCN+EdgeBoxes[14]		scale-aware TSR (ours)	
	Prec.(%)	Rec.(%)	Prec.(%)	Rec.(%)	Prec.(%)	Rec.(%)	Prec.(%)	Rec.(%)	Prec.(%)	Rec.(%)	Prec.(%)	Rec.(%)
PEDESTRAIN CROSSING	96.03	91.77	98.52	93.45	87.9	87.2	94.21	97.44	100	95.20	100	99.15
PASS RIGHT SIDE	100	95.33	100	97.53	93.8	93.8	94.44	96.23	95.3	93.8	100	100
NO STOPPING NO STANDING	97.14	77.27	99.20	81.46	66.8	71.7	85.71	54.55	100	75.0	100	100
50 SIGN	100	76.12	100	80.56	100	100	100	100	100	100	100	100
PRIORITY ROAD	98.66	47.76	97.89	79.68	95.7	97.8	96.51	98.81	100	98.9	98.77	95.24
GIVE WAY	59.26	47.76	71.50	52.39	79.4	90	100	85.71	96.7	96.7	100	96.43
70 SIGN	-	-	-	-	92.6	86.2	100	100	100	100	100	100
80 SIGN	-	-	-	-	100	77.3	100	100	94.4	77.3	100	95.24
100 SIGN	-	-	-	-	100	100	100	73.68	90.5	100	100	94.74
NO PARKING	-	-	-	-	96.3	68.4	100	76.47	100	92.1	100	85.29
Average (the first 6 classes)	91.84	77.08	94.52	80.85	90.08	87.27	95.15	88.79	98.67	93.27	99.79	98.47
Average (all)	-	-	-	-	91.25	87.24	97.09	88.29	97.69	92.90	99.88	96.61

Table 2: Accuracy comparison on STSD based on standard evaluation benchmark

Methods	Region Proposal		Classification		Overall		Accuracy	
	Time(s/img)	Parameters(million)	Time(s/img)	Parameters(million)	Time(s/img)	Parameters(million)	Prec.(%)	Rec.(%)
FCN+EdgeBoxes[14]	0.578	14.7	0.052	53.4	≈0.63	68.1	97.69	92.90
Faster R-CNN [14]	0.423	17.1	0.041	119.7	0.511	136.8	97.09	88.29
scale-aware TSR	0.226	0.5	0.102	1.8	0.413	2.4	99.88	96.61
scale-aware TSR (64 proposals)	0.226	0.5	0.056s	1.8	0.356	2.4	99.88	96.52

Table 3: Comparison of running time and model complexity.

training data. After the convergence of the network training, we use the model to find the wrongly classified samples and add them to the training set to further optimize the network. Most of the hyper-parameters are the same with those in training the proposal network, except that the batch size is 128, the learning rate is decreased by a factor of 5 after 20k iterations and the training ends after 100k iterations.

Post Processing: In the testing stage, we feed the bounding boxes generated by the DMS-net to the fusion-net. Then we perform non-maximum suppression over all the region proposals according to their classification scores. Finally, we adopt the bounding boxes voting trick [[14](#)] to further boost the location accuracy.

4 Experiments

The proposed scale-aware TSR system is implemented in the framework of Caffe [[14](#)], and run on a machine with a 4-core CPU@2.7GHz, 16G RAM, and a NVIDIA K40 GPU. We evaluate our method on STSD [[14](#)], which consists of more than 2,000 annotated images. We follow the experimental configurations in [[14](#)] to split the training and testing data.

4.1 Evaluation on Standard STSD Benchmark

We compare our system with state-of-the-art methods [[6](#), [14](#), [17](#), [30](#), [24](#)] based on the standard evaluation benchmark proposed in [[17](#)], where only traffic signs labeled as 'visible' and larger than 50×50 are considered. The performances of the methods are measured by precision and recall separately, where results having IoU with ground truth higher than 0.5 are treated as true positive.

Comparison on Accuracy: Table 2 gives precisions and recall rates of FDs [[17](#)], Adaboost+SVR [[6](#)], R-CNN [[14](#)], Faster R-CNN [[30](#)], FCN+EdgeBoxes [[14](#)] and our scale-aware TSR. For Faster R-CNN, we use VGG16 as its backbone network and 960×1280 as input size. All the training and testing settings are the same with the original paper [[30](#)]. For

all the experiments except those specifically stated ones, we truncate 128 proposals generated by DMS-net. As shown in table 2, our method achieves an average precision of 99.79% and an average recall of 98.33% on the first 6 classes, which are much better than those of any other methods. Most notably, the improvement on the recall rate, 5.06 points higher than the best method [44], is significant for traffic sign recognition. We argue that it mainly gains from the dual multi-scale design in the DMS-net, which will be verified in the next section. For all the classes, we also have the best performance (99.88% precision and 96.61% recall rate). In comparison with the baseline method [44], we gain 2.19% and 3.71% improvement on precision and recall rate, respectively. Moreover, we observe that the classification of all types of speed signs generated by our proposal network are 100% correct, due to the multi-scale feature fusion structure in the fusion-net enabling fine-grained recognition.

Comparison on Efficiency: Fast computation and low model complexity are essential for practical applications. We take the top 3 methods (Faster R-CNN [50], FCN+EdgeBoxes [44] and ours) in Table 2 for model complexity and time cost comparison. Due to the lightweight consideration in system design, the number of parameters in our entire system is around $57\times$ lower than that in Faster R-CNN and around $28\times$ lower than that in [44]. Therefore, our system is better suited to automobiles with on-board hardware. In order to test the time costs, we run the above three methods with their default settings on our machine whose configuration was presented earlier. Since the overall code of FCN+EdgeBoxes [44] is not public, we reproduce their work and estimate their time costs on our machine. The time may have a minor difference with that presented in their paper due to different machines. We show the speeds of each stage and the overall systems in Table 3. Although our classification is slower than that of the others, our whole system is the fastest one (0.413s in total running time). Furthermore, We also attempt to reduce the region proposals from 128 to 64. Our method gains 0.057s speedup, while it only drops 0.09% recall rate.

4.2 Ablation Study

We conduct a deep analysis on the STSD dataset and evaluation metrics. Figure 3 reports the data distribution of the testing dataset. According to this distribution, most of the traffic signs in the STSD testing dataset are smaller than 50×50 pixels, only 30.8% traffic signs satisfy the requirements of the evaluation benchmark in [44]. In other words, the standard STSD benchmark ignores small-scale traffic signs which are widespread in the real world. Detecting small objects is always a challenge, not only in traffic sign recognition [45] but also in generic object detection tasks [46]. Motivated by the mainstream evaluation metrics in MS COCO [47], we redesign a new evaluation method on the STSD dataset, which takes the wide-range traffic signs into account. In this evaluation method, we incorporate all size of traffic signs into the evaluation and split them into three different scales according to their areas. As shown in Figure 3, $area < 32^2$, $32^2 < area < 96^2$, and $area > 96^2$ denote small, medium and large scale, respectively. In this way, it is possible to evaluate TSR methods based on not only their overall performances but also their multi-scale performances. Moreover, we adopt Average Precision (AP), the area under the precision-recall curve, to measure

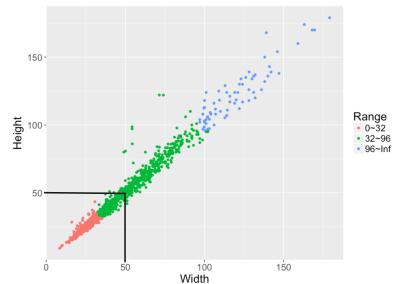


Figure 3: Distribution of the traffic signs in STSD testing dataset.

Methods	AR@50	AR@100	AR@300	AR@Small	AR@Medium	AR@Large
Selective Search [58]	4.8	10.7	23.4	1.1	12.3	33.4
EdgeBoxes [46]	36.9	41.0	47.4	22.0	48.3	67.9
RPN [30]	66.9	67.0	67.2	57.6	71.4	72.1
DMS-net (w/o branch1)	52.5	52.8	53.0	5.7	74.6	77.2
DMS-net (w/o branch2)	67.0	71.3	72.3	61.4	75.8	75.3
DMS-net	69.8	71.5	72.3	62.2	75.6	76.7

Table 4: Average Recall (AR) analysis obtained with different proposal numbers and scales. AR for small, medium and large objects are computed for 100 proposals.

class-wise performances. We utilize Average Recall (AR) metric which is an averaged recall over different IoU thresholds (0.5, 0.55, \sim 0.95) to evaluate region proposal methods.

Analysis on DMS-net. To verify the scale-awareness of our DMS-net, we compare it against three well-known state-of-the-art methods for generic object proposals, including Selective Search [58], EdgeBoxes [46] and RPN [30]. Table 4 reports the AR values at different proposals and different scales. We can see that our DMS-net is significantly better than the others on all metrics. Our DMS-net also exceeds non-learning methods (Selective Search and EdgeBoxes) by a large margin. Compared with RPN method, we gain 4.6% improvement on both the small-scale and large-scale traffic signs. This proves that our method is effective in detecting multi-scale traffic signs. We also conduct an ablation study by removing the first or second branch but keeping the same number of scales at 10 to verify the effectiveness of the proposed dual multi-scale structure. We observe that if we remove the first branch, the performance on detecting small traffic signs decrease dramatically. The DMS-net without the second branch also has inferior ARs on small and large traffic signs, even though it has a minor increase (0.2%) relative to our entire DMS-net on AR at medium ones. Moreover, We observe that our DMS-net limitedly benefits from the increase in the number of proposals. This means that our method can achieve satisfying results even with few proposals, thereby saving computing time in the later classification task.

Analysis on fusion-net. In order to evaluate our fusion-net, we compare it with the classifier in our baseline method [42]. For fair comparison, we use the same region proposal method, our DMS-net, to generate proposals for the two classifiers. As we can see from Table 5, our fusion-net performs better in most categories of traffic signs and achieves 1.87% mAP higher than the baseline. Moreover, we compare our entire system (DMS-net + fusion-net) with Faster R-CNN. The results in Table 5 show that our scale-aware TSR outperforms Faster R-CNN by a large margin, 11.67% mAP. We further replace their classifier with our fusion-net, it gains around 10% mAP improvement. This also proves the effectiveness of our fusion-net. For the multi-scale evaluation, our scale-aware TSR obtains high AP scores at different scales. Especially, it achieves 98.1% AP at medium-scale traffic signs.











Methods	AP (small)	AP (medium)	AP (large)	mAP										
DMS-net+ fusion-net	85.19	98.10	96.29	94.67	96.74	98.97	86.67	96.55	95.75	90.44	98.91	95.45	97.04	90.11
DMS-net+[42]classifier	82.42	97.10	94.15	92.80	96.73	98.94	86.67	96.20	94.62	94.34	98.91	81.65	92.90	87.70
Faster R-CNN[42]	52.42	93.96	97.72	83.00	86.54	88.83	59.32	81.70	82.48	88.24	91.66	87.25	75.64	88.35
RPN[30]+ fusion-net	86.63	97.30	89.64	93.41	94.65	97.92	84.56	95.98	90.33	94.82	97.82	95.45	95.42	87.13

Table 5: Average Precision (AP) on all the data and those of different scales and categories.

5 Conclusion

The paper presented a traffic sign recognition system with performance beyond state-of-the-art methods. The system adopted an FCN with a dual multi-scale CNN architecture to detect traffic signs of different scales, and a concise CNN structure to fuse multi-scale features for classification. Besides, multiple existing effective strategies and modules, e.g. OHEM and Inception, were introduced into the system. The whole system and its modules were evaluated thoroughly on STSD. The system was experimentally proved to be more accurate and faster than state-of-the-art methods. Moreover, the system is more lightweight than the others, thereby more suitable for automobiles with on-board hardware. Future work will be focused on strengthening the system for more challenging tasks, e.g. recognizing traffic signs from images captured in fog weather.

6 Acknowledgement

We thank NVIDIA for the donation of a Tesla K40 GPU. This research is also supported by Beijing Municipal Natural Science Foundation (4152006), Scientific Research Project of Beijing Educational Committee (KM201510005015), and National Natural Science Foundation of China (61003105, 61370113).

References

- [1] Hamed Habibi Aghdam, Elnaz Jahani Heravi, and Domenec Puig. A practical approach for detection and classification of traffic signs using convolutional neural networks. *Robotics and Autonomous Systems*, 84:97–112, 2016.
- [2] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *CVPR*, pages 2874–2883, 2016.
- [3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *ECCV*, pages 354–370, 2016.
- [5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [6] Tao Chen and Shijian Lu. Accurate and efficient traffic sign detection using discriminative adaboost and support vector regression. *IEEE Transactions on Vehicular Technology*, 65(6):4006–4015, 2016.
- [7] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. A committee of neural networks for traffic sign classification. In *IJCNN*, pages 1918–1921, 2011.
- [8] Dan Cireşan, Ueli Meier, Jonathan Masci, and Jürgen Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012.

- [9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.
- [10] Richard O Duda and Peter E Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [11] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, pages 1–8, 2008.
- [12] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*, pages 1134–1142, 2015.
- [13] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015.
- [14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.
- [15] Mrinal Haloi. Traffic sign classification using deep inception based convolutional networks. *arXiv preprint arXiv:1511.02992*, 2015.
- [16] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312, 2014.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, pages 346–361, 2014.
- [18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.
- [19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.
- [20] Junqi Jin, Kun Fu, and Changshui Zhang. Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 15(5):1991–2000, 2014.
- [21] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: towards accurate region proposal generation and joint object detection. In *CVPR*, pages 845–853, 2016.
- [22] Fredrik Larsson and Michael Felsberg. Using fourier descriptors and spatial models for traffic sign recognition. In *Scandinavian Conference on Image Analysis*, pages 238–249, 2011.
- [23] Ming Liang, Mingyi Yuan, Xiaolin Hu, Jianmin Li, and Huaping Liu. Traffic sign detection by roi extraction and histogram features-based recognition. In *IJCNN*, pages 1–8, 2013.
- [24] Ming Liang, Mingyi Yuan, Xiaolin Hu, Jianmin Li, and Huaping Liu. Traffic sign detection by roi extraction and histogram features-based recognition. In *IJCNN*, pages 1–8, 2013.

- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.
- [26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.
- [27] Saturnino Maldonado-Bascon, Sergio Lafuente-Arroyo, Pedro Gil-Jimenez, Hilario Gomez-Moreno, and Francisco López-Ferreras. Road-sign detection and recognition based on support vector machines. *IEEE transactions on intelligent transportation systems*, 8(2):264–278, 2007.
- [28] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.
- [29] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, pages 82–90, 2014.
- [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [31] Samuele Salti, Alioscia Petrelli, Federico Tombari, Nicola Fioraio, and Luigi Di Stefano. A traffic sign detection pipeline based on interest region extraction. In *IJCNN*, pages 1–7, 2013.
- [32] Pierre Sermanet and Yann LeCun. Traffic sign recognition with multi-scale convolutional networks. In *IJCNN*, pages 2809–2813, 2011.
- [33] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.
- [34] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016.
- [35] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *IJCNN*, pages 1453–1460, 2011.
- [36] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks*, 32:323–332, 2012.
- [37] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.
- [38] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

- [39] Dongdong Wang, Xinwen Hou, Jiawei Xu, Shigang Yue, and Cheng-Lin Liu. Traffic sign detection using a cascade method with fast feature extraction and saliency test. *IEEE Transactions on Intelligent Transportation Systems*, (99):1–13, 2017.
- [40] Gangyi Wang, Guanghui Ren, Zhilu Wu, Yaqin Zhao, and Lihui Jiang. A robust, coarse-to-fine traffic sign detection method. In *IJCNN*, pages 1–5, 2013.
- [41] Yi Yang, Hengliang Luo, Huarong Xu, and Fuchao Wu. Towards real-time traffic sign detection and classification. *IEEE Transactions on Intelligent Transportation Systems*, 17(7):2022–2031, 2016.
- [42] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.
- [43] Fatin Zaklouta, Bogdan Stanciulescu, and Omar Hamdoun. Traffic sign classification using kd trees and random forests. In *IJCNN*, pages 2151–2155, 2011.
- [44] Yingying Zhu, Chengquan Zhang, Duoyou Zhou, Xinggang Wang, Xiang Bai, and Wenyu Liu. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, 214:758–766, 2016.
- [45] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *CVPR*, pages 2110–2118, 2016.
- [46] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014.