Faster R-CNN for Small Traffic Sign Detection

Zhuo Zhang[1], Xiaolong Zhou[1], Sixian Chan[1], Shengyong Chen[1], and Honghai Liu[2]

[1] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
[2] School of Computing, University of Portsmouth, Portsmouth, UK

Abstract. Traffic sign detection is essential in autonomous driving. It is challenging especially when large proportion of instance to be detected are in small size. Directly applying state-of-the-art object detection algorithm Faster R-CNN for small traffic sign detection renders unsatisfactory detection rate, while a higher accuracy will be performed if the input images are upsampled. In this paper, we first investigate Faster R-CNN's network architecture, and regard its weak performance on small instances as improper receptive field. Then we augment its architecture according to receptive field with a higher accuracy achieved and no obvious incremental computational cost. Experiments are conducted to validate the effectiveness of proposed method and give an comparison to the state-of-the-art detection algorithms on both accuracy and computational cost. The experimental results demonstrate an improved detection accuracy and an competitive computing speed of the proposed method.

Keywords: Traffic Sign Detection, Convolutional Neural Network, Receptive Field

1 Introduction

Traffic signs such as traffic lights and road signs play an important role in driving scene. They are designed to inform drivers of the current traffic situation, and their location information bridges the detection and recognition procedures. In driving assistant system, finding their location and determine their category can help drivers further reduce accident happening rate. We are interested in determining traffic signs' positions, and in this paper we focus on small traffic signs detection, since most instances only occupy a small relative area. Being a concrete case of object detection, we consider applying existing object detection algorithm on this task.

In recent years, deep Convolutional Neural Network (CNN) based methods on object detection have a fairly good performance. They use large amounts of data for training, or take advantage of transfer learing, and with the help of GPU computation, high accuracy is gained.

Faster R-CNN[16] is one of the state-of-the-art object detection algorithms. It employs convolution nework to generate region proposals and further refine their locations and categories. This approach achieves impressive performance on the PASCAL VOC[6] benchmark. The dataset we use is released from CCF BDCI 2016 competition. As shown in Fig. 1, this dataset contains large amounts of traffic signs that cover a small region of whole image, which is very different from the VOC2007 and pushes us in a dilemma when applying Faster R-CNN to achieve a high accuracy. One way to solve this problem is to upsample input images such that the effective objects' sizes would be similar to those in VOC2007. However, this will lead to a larger computational cost. In contrast, if we use the same resize configuration on input images or don't resize for fairly small objects, the computational cost would be better but the accuracy would be undesirable.

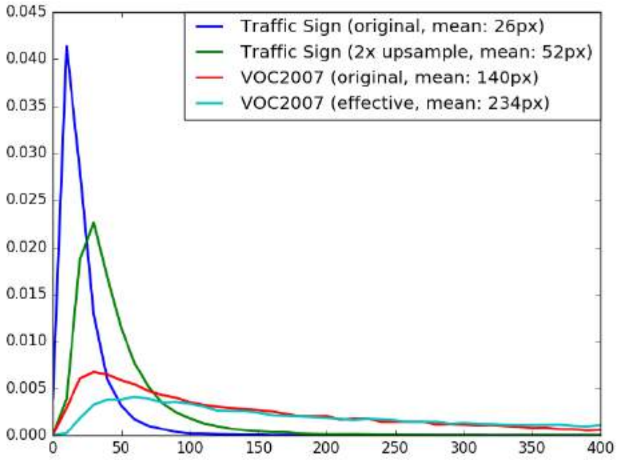

Fig. 1. Fig. 1 Square area distribution of VOC2007 dataset and small traffic sign dataset. Obviously, even with 2x upsampling, traffic signs are still smaller than VOC2007 objects, which make it harder to detect.

In this paper, we try to propose an effective method to balance the computational cost and detection accuracy. Actually, upsampling input images is to decrease the network's receptive field. To decrease the receptive field, we can modify the network architecture instead of resizing the input images. For example, we can change convolution filer's size and stride, or change feature map resolution. If there is no input images resizing, we could gain fairly good performance by modifying the network architecture to keep a receptive field similar to upsampling scheme's. Comparing to the strategy of no input images resizing and original network, the accuracy would be guaranteed. Comparing to input upsampling with orignal network, the accuracy may be slightly lower, but the computational speed in both training and inference is fast and model size is also decreased. Meanwhile, we replace Faster R-CNN detection subnetwork's fully connection layers to convolution layers, which will further reduce the model size.

In this paper, we focus on proposing an improved Region Proposal Network (RPN) network architecture (RF + DilatedConv) to apply the Faster R-CNN on the challenging task of small traffic sign detection. Although the proposed method follows the Faster R-CNN, there are at least three major differences.

(1) We extend the Faster R-CNN in a new application to detect the challenging (small) traffic signs.

(2) We propose a receptive field guided Region Proposal Network (RPN) which boosts proposal quality.

(3) In the R-CNN detection subnetwork, we use fully convolution network to replace fully connected network, which keeps the accuracy and reduces the model size.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in both traffic sign and generic object detection. Section 3 presents analysis on RPN, gives computation basics of receptive field, and details the modification of the RPN architecture based on receptive field. In section 4, extensive experiments are reported, presenting the correctness of the proposed method and competitive computing speed. Section 5 concludes this work.

2 Related Work

Early traffic sign detections are mainly in ideal conditions, where target object-s occupy a large or medium proportion of the image. Most of them are clear and less occlusion. Researchers combine color and geometry characteristics to tackle this problem. For example, Fleyeh [7] detects traffic sign based on the color segmentation. Xu et al. [14] take advantage of shape symmetry for judging traffic signs. Later, more practical benchmark GTSDB [11] is proposed. Encouraged by the success of HOG feature and SVM classifier in human and generic object detection[4], this algorithm and its variants renders good accuracy on corresponding datasets[19, 11, 5]. However, the GTSDB benchmark is still not representative of that encountered in real tasks.

After Convolutional Neural Network (CNN) is rekindled in image classification [12, 17], many CNN based object detection algorithms are proposed [16, 9, 10, 8, 15, 13, 3], essentially based on the rich representation of deep layers and additional adaptive subnetworks. Among them, RCNN[9], SPPNET[10], and Fast RCNN[8] first use existing region proposal method to generate candidate region-s, then a DCNN model learns feature representation from all of them and gives a trained model, which is used for candidate region classification in inference time. All of them consists of two stage separate modules. Differ from that, Faster R-CNN[16], YOLO[15], SSD[13], and R-FCN[3] use only one network for the whole object detection task, thus the features are all learned rather than designed or partly designed via manually designing. These DCNN based algorithms can also be divided into two types: Faster R-CNN and R-FCN. Similarly, they also have two stages: learn a region proposal generator and then classify these proposals via following network with predicted location refined. While on the other hand, YOLO[15] and SSD[13] consist of only one single network, and they generate predicated object region with class labels directly.

Based on the work of Faster R-CNN, some algorithms are proposed for concrete purposed detection task. MSCNN[1] extends RPN[16] to multi-scale so that receptive field can match objects of different scales. RPN-BF[20] adopts RPN to generate high-resolution feature map to detect small pedestrian instances. Both MSCNN and RPN-BF deal with small object detection, and it is equivalent to gain smaller receptive field by obtaining high-resolution RPN last layer feature map. However, they conclude the bad performance of RPN or Faster R-CNN on these objects to improper receptive field, and their improvements are based on this discovery. Our method is based on them, but gives a formal receptive field calculation and thus provides a helpful reference for small object detection.

3 Proposed Method

3.1 Analysis on RPN

Faster-RCNN is a two-stage object detector, consisting of RPN and Fast RCNN subnetworks. RPN generates candidate object regions and Fast RCNN network classifies them and refines their locations. Region proposals' qualities determine the final detection performance on a large scale. They are found in low quality in the task of small traffic signs' detection. This subsection explores the cause for this deficiency and proposes three improvement requirements.

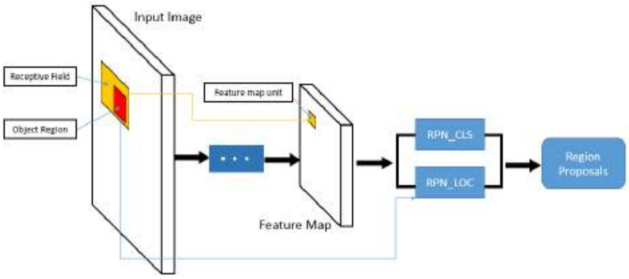

Fig. 2. Architecture of RPN.

The architecture of RPN is shown in Fig. 2. It has several convolution and pooling layers, followed by data manipulation and loss layers, which joins both classification and location regression tasks and generates region proposals finally. Feeding images to the network, feature maps are generated after each convolution and pooling layer. The resolution of the last layer's feature maps is much smaller than input images. Each unit on the last feature map depends on a region of pixels of the input image, i.e. its receptive field (NRF). The size of NRF should be close to the target objects, since oversized NRF fuses too much background information and makes features less discriminative. Decreasing the network's receptive field is the first requirement in our experiment.

In RPN, it generates a set of windows with specified scale and aspect ratios. This set of windows share the same center, and the center can be any receptive field's center, thus all these windows consist of the network's reference windows. An anchor is labeled positive if it has an Intersection-over-Union (IoU) greater than 0.5 with one ground truth box, and otherwise negative. Each anchor's corresponding unit on the last feature map for classification task is with the same label. For small object detection, oversized reference windows generate less positive samples, which harms the classification and region proposal's quality. Thus, shrinking reference windows' size is the second improvement requirement. Note that the RoI pooling layer bridges the gap between RPN and Fast RCNN, candidate objects' corresponding region on the last feature map serves as the input of Fast RCNN detection sub-network. Small target objects give birth to small region proposals, thus corresponding feature map region has low-resolution. This gives rise to less discriminative features because of collapsing bins, and thus degrades the downstream classifier. Therefore, the third improvement requirement comes that candidate objects' corresponding feature map region should be big enough for RoI pooling.

We can use simple tricks to meet the three requirements. By dropping out layer, modifying filter's stride or size, and using dilated convolution, the receptive field gets smaller. By decreasing anchor's base size, the reference window shrinks. By decreasing filter's stride, the feature map's resolution becomes larger. We give the fundamental for receptive field's calculation in the next subsection.

3.2 Calculation of Receptive Field

In convolutional networks, each unit's value depends on a region of the input. That region in the input is used as the receptive field for that unit. The input layer filters' size and stride determine how big the receptive field can be. Actually, each unit in the input region also depends on another region of the more previous layers (if any). We generalize this concept by replacing one unit with multiple units that consist of a square region. Thus, a unit's receptive field can be determined layer by layer. Since RPN receptive field is employed to match ground truth objects, we care how big the receptive field is on the input image. We treat the input image as the 0th layer, and assume that the network consists of n convolution layers (including pooling layer). Let the receptive filed of one unit of the last convolution layer's feature map be RF, it can be calculated via:

$$RF = F_n(1) \tag{1}$$

$$F_n(m) = F_{n-1}(k_n + (m-1)s_n) \tag{2}$$

$$F_0(m) = m \tag{3}$$

where $F_n(m)$ denotes the receptive field of m units on layer n, k_n and s_n represent the size and stride of n^{th} layer's kernel respectively. The 0^{th} layer is the input image itself.

3.3 Dilated Convolution

Dilated convolution is a general form of convolution operation[18]. It sums activation of signals with equal distances, which looks like using a filter with holes. We call it p-dilated convolution when the distance is p, and the equal distance of nearby pixels is $p+1$. Obviously, when the distance equals to 1, it is the ordinary convolution operation. For 2D images, the p-dilated convolution operation can be defined as:

$$(W *_p I)(x,y) = \sum_{s=-a}^{a} \sum_{t=-a}^{a} W(s,t)I(x-ps, y-pt) \tag{4}$$

where I represents the image, W is the convolutional filter, (x,y) is the center point for filter and the filter's length is $2a+1$, $*_p$ means the distance of nearby pixels selected for computation is p (on horizontal or vertical orientation).

Comparing to vanilla convolution operation, p-dilated convolution generates the sized feature map but with larger receptive field. Specifically, for p-dilated convolution, the filter's length is updated as:

$$k' = pk - 1 \tag{5}$$

With the updated kernel size k', the receptive field increases. For example, assuming all the kernels' length is 3 and stride is 1, the first layer and second layer do 1-dilated and 2-dilated convolution respectively, then the second feature maps' each unit's receptive field size is 7 instead of 5.

4 Experiments

4.1 Dataset and algorithm parameters setting

In order to verity the effectiveness of our proposed method, we use same set of training and test data and choose 3 methods to compare: Faster R-CNN, our proposed method and SSD, and evaluated their performance with Average Precision (AP) at intersection and union area overlap ratio of 0.5.

The dataset is from the preliminary contest of CCF BDCI 2016 Traffic sign detection in self-driving scenario, which contains 4000 images with 720 height and 1280 width. These images are picked up from taxi's driving recorder and vary in illumination and angles. About 28000 traffic signs (mainly traffic lights and road signs) are labeled. Each image contains about 7 sign instances on average. We randomly choose 3000 images for training, the rest for validation.

Five concrete methods are performed on this dataset with corresponding Average Precision evaluated for comparison.

Roughly we compare three algorithms: Faster R-CNN, our proposed Receptive Field net (RFnet), and SSD. For the purpose of comparing different receptive field, Faster R-CNN is with original size image and 2x up-sampled image as input. Our RFnet is designed to compete the latter in accuracy, and being a common RFnet and a RFnet with dilated convolution. We choose ZF-net[2] to fine-tune them on, since this backbone network consumes a video ram that a NVIDIA GTX970 GPU can handle, especially for 2x up-sampled Faster R-CNN. Being competitive in both accuracy and speed in common object detection, we also train and test a SSD model with VGG16 backbone network, to compare with Faster R-CNN and our RFnet on small traffic sign detection. Hence, there are 5 experiment schemes in total. Scheme names and corresponding measurement are listed on Table 1, and Precision-Recall curve and AP value plotted on Fig3.

4.2 Experiment schemes and results

For Faster R-CNN and our RFnet, we fine-tune them on ZF-net with 70000 training iteration. For SSD we fine-tune it on VGG16 with 50000 training iteration. All these schemes are trained on 3000 images and evaluated on 1000 images with AP@0.5 as the measurement. Their performance is plotted on Fig 3. For a fixed recall, the higher precision the better accuracy. For different receptive field, original input images and 2x up-sampled images are fed into Faster R-CNN, denoted as FRCNN-ZF-1x-input and FRCNN-ZF-2x-input, which generate receptive fields of 171 and 85 respectively. Since the traffic sign detection dataset contains large number of small size instance (see Fig 1), and Faster R-CNN expects not too small object sizes, it is not surprising that FRCNN-ZF-2x-input performs better than FRCNN-ZF-1x-input.

Table 1. Details of experiment schemes.

Scheme	Receptive Field	LCS resolution	Model size	Speed	AP (70000 iteration)
FRCNN-ZF-1x-input	171	16	255M	0.14s	30.7%
FRCNN-ZF-2x-input	85	16	255M	0.37s	44.6%
SSD-VGG16-512	-	-	91M	0.13s	41.9%
RFnet-ZF(ours)	83	8	34M	0.41s	50.0%
RFnet-ZF-dilated(ours)	85	8	29M	0.39s	48.2%

Our RFnet also gives a smaller receptive field that nearly the same as FRCNN-ZF-2x-input, without input image re-scaling. Based on Faster R-CNN network architecture, we decrease the last shared convolution layer's receptive field by means of altering convolution or pooling layer's kernel size and stride, and dilated convolution trick if possible. We give two concrete networks. One is RFnet-ZF, which drops conv5 layer and decreases the second pooling layer's size from 3 to 2 and stride from 2 to 1. The other is RF-net-dilated, which drops conv4 and conv5 and decreases pool2's size from 3 to 2, stride from 2 to 1, and pool1's size from 3 to 4. Both of them shrink the receptive field to nearly half of before, without re-scaling of input. The anchor window's basic sizes also decrease, for the sake of better IoU of anchor box and ground truth box. In experiments, we use 2, 4, 8 as basic sizes. Since our RFnet is based on Faster R-CNN, we also perform 70000 iterations of back-propagation. As illustrated in Fig. 3, our RFnet performs the best, which achieves 50% AP value, and RFnet-dilated performs the second best, which achieves 48.2% AP. Comparing to FRCNN-ZF-1x-input, anchor boxes and RPN training samples are generated on larger resolution feature map of last shared convolution layer (denote as LSC resolution), while they have same input sizes. Therefore, RFnet uses half receptive field and outperforms Faster R-CNN a large margin. Comparing with FRCNN-ZF-2x-input, our RFnets have better accuracy while they keep the same receptive field. This is mainly due to the modified anchor size, which makes the network consumes longer time both in training and test phase. As illustrated on Table 1, RFnet-ZF is with a 5.4% higher accuracy than FRCNN-ZF-2x-input, with the cost of 0.04s longer for each image inference.

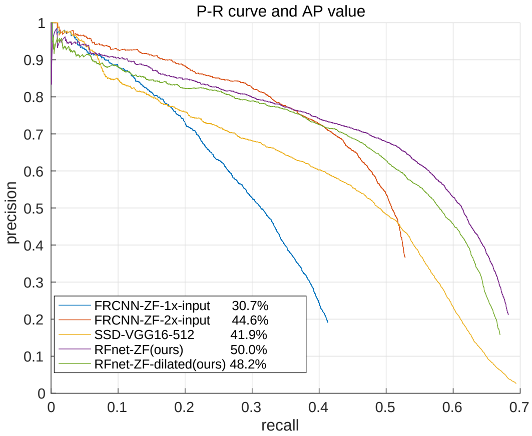

Fig. 3. Performance on the traffic sign detection dataset

We also do comparison with other state-of-the-art CNN based object detection algorithms to show the superior performance of the proposed method. We choose SSD, the representative of only one stage conv-net based method, known for its accuracy and fast inference speed. Since the limitation of time, we only pick SSD with VGG16 as backbone network and input image size of 512. Ob-

Table 2. Details of experiment schemes.

scheme	image scale	anchor base size	changes on architecture	NRF	AP (70000 iteration)
scheme1	1x	16	no change	171	31.3%
scheme2	2x	16	no chnage	85	45.4%
scheme3	1x	8	remove conv5 pool2.k:3→2 pool2.s:2→1 feat_stride:16→8 spatial_scale:$\frac{1}{16} \rightarrow \frac{1}{8}$	83	44.2%
scheme4	1x	8	remove conv4 remove conv5 pool2.k:3→2 pool2.s:2→1 pool1.k:3→4 conv3:2-dilated feat_stride:16→8 spatial_scale:$\frac{1}{16} \rightarrow \frac{1}{8}$	85	40.0%
scheme5	1x	8	remove conv4 remove conv5 pool2.k:3→ 2 pool2.s:2→1 pool1.k:3→2 conv3:2-dilated feat_stride:16→8 spatial_scale:$\frac{1}{16} \rightarrow \frac{1}{8}$	81	41.0%

viously, this scheme should have better accuracy than ZF-net based SSD. We use batch size of 4 and learing rate of 0.00025, and consider the batch size of Faster R-CNN and RFnet as 1. We perform 50000 iterations for SSD. We shrink the input to 512×512 and denote this net as SSD-VGG16-512. For accuracy, as illustrated on Fig 3, SSD-VGG16-512 has higher precision than FRCNN-ZF-1x-input at most time, while lower than FRCNN-ZF-2x-input at low recall rate, but higher precision for higher recall. Thus, SSD is with 41.9% AP, smaller but near the performance of FRCNN-ZF-2x-input with 44.6% AP. Considering the number of parameters, i.e. the model size, SSD-VGG16-512 is 91M, which is only 35% of FRCNN with ZFnet, and also runs faster than FRCNN, SSD is competitive to it. Meanwhile, our RFnets surpass SSD on nearly all possible recall rate, and reaches 48.2% and 50.0% AP, which is much higher than SSD. This accuracy difference shows the effectiveness of our proposed RFnet. Since RFnet also use convolution layers to replace fully connected layers, the model size also shrinks. Luckily, RFnets use around 1/3 number of parameters than SSD and have better accuracy. The shortcoming of RFnets to SSD is the inference speed. Our RFnets consumes 2 more times than SSD, with nearly 0.4s for each image.

Fig. 4 demonstrates the detection results of five algorithms. From the first row(r1) to the last row(r5), the detection results are obtained by the FRCNN-ZF-1x-input, FRCNN-ZF-2x-input, SSD-VGG16-512, RFnet-ZF, and RFnet-ZF-dilated, respectively. The results show that the proposed RFnet detects more correct small traffic signs.

5 Conclusion

In this paper, we have presented a simple but effective baseline that adopted Faster-RCNN's network architecture for small traffic sign detection. An analysis on the RPN network has showed that there needs a proper match among receptive field, reference window and traffic sign. We have proposed to modify the convolution network's architecture. The specific method included increasing anchors' density and feature maps' resolution, and decreasing receptive field and reference window size. These improvements brought accurate candidate regions, and kept smoothing connection with the following detection sub-network. The proposed method has gained remarkable performance boost. Meanwhile, the network generated smaller sized model due to less layers in use, and ran faster at test stage. The experimental results demonstrated the good performance of the proposed method in detecting small traffic signs, which could be further employed to multi-scale object detection.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (61403342, U1509207, 61325019, 61673329, 61603341).

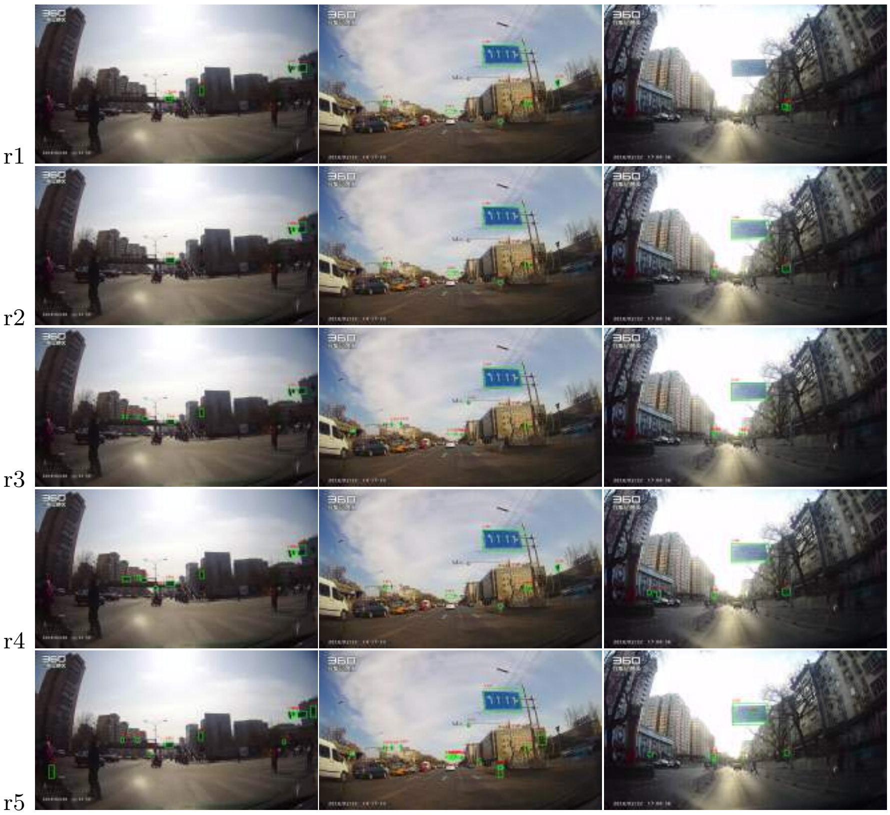

Fig. 4. Detection results of five detection algorithms on same set of images.

References

1. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision. pp. 354–370. Springer (2016)
2. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
3. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems. pp. 379–387 (2016)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 1, pp. 886–893. IEEE (2005)
5. El Margae, S., Sanae, B., Mounir, A.K., Youssef, F.: Traffic sign recognition based on multi-block lbp features using svm with normalization. In: Intelligent Systems: Theories and Applications (SITA-14), 2014 9th International Conference on. pp. 1–7. IEEE (2014)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision 88(2), 303–338 (2010)
7. Fleyeh, H.: Shadow and highlight invariant colour segmentation algorithm for traffic signs. In: Cybernetics and Intelligent Systems, 2006 IEEE Conference on. pp. 1–7. IEEE (2006)
8. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 580–587 (2014)
10. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: European Conference on Computer Vision. pp. 346–361. Springer (2014)
11. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: Neural Networks (IJCNN), The 2013 International Joint Conference on. pp. 1–8. IEEE (2013)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European Conference on Computer Vision. pp. 21–37. Springer (2016)
14. Qingsong, X., Juan, S., Tiantian, L.: A detection and recognition method for prohibition traffic signs. In: Image Analysis and Signal Processing (IASP), 2010 International Conference on. pp. 583–586. IEEE (2010)
15. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision 115(3), 211–252 (2015)
18. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: ICLR (2016)
19. Zaklouta, F., Stanciulescu, B.: Real-time traffic sign recognition using spatially weighted hog trees. In: Advanced Robotics (ICAR), 2011 15th International Conference on. pp. 61–66. IEEE (2011)
20. Zhang, L., Lin, L., Liang, X., He, K.: Is Faster R-CNN doing well for pedestrian detection? In: European Conference on Computer Vision. pp. 443–457. Springer (2016)