ORIGINAL ARTICLE

INTERNATIONAL JOURNAL OF CONVERGENCE IN HEALTHCARE

IJCIH
Converging Healthcare & Technology

Published by
IJCIH & Pratyaksh Medicare LLP

www.ijcih.com

Effectiveness of Focal Loss for Traffic Sign Detection Using Deep Neural Networks

Deepika[1], Sharda Vashisth[2], Prabha Sharma[3]

[1]Research Scholar, Department of Multi-Disciplinary Engineering, The NorthCap University, Gurugram, India, [2]Professor, Department of Multi-Disciplinary Engineering, The NorthCap University, Gurugram, India, [3]Professor, Department of Applied Sciences, The NorthCap University, Gurugram, India, Deptt. of Multi-Disciplinary Engineering, The NorthCap University, Haryana India

Abstract

Recent developments in autonomous driving have created a great demand for precise and computationally effective traffic sign-detecting systems. By assisting drivers and assuring their safety, such technology can lessen traffic accidents and fatalities. However, to make such a system deployable, a few crucial accuracy and processing performance problems must be resolved. Real-time performance is sometimes regarded as a must for such an application. RetinaNet, a focal loss-based single-stage object detector, is employed to strike a compromise between accuracy and processing speed concerning the most cutting-edge object detectors. The detector is suitable for traffic sign identification since it was developed to overcome the class imbalance problem that the single-stage detector had. (TSD). The efficiency of the detector was evaluated by combining feature extractors like ResNet-50 and ResNet-101 on two openly available TSD benchmark datasets. Various metrics like memory allocation, mean average precision (mAP), running time, amount of floating-point operations, and model parameters are taken into consideration. Evaluation of the detector on several datasets is required to examine the variance in the performance, and the RetinaNet model is the fastest and best model in terms of memory usage, making it the ideal option for the deployment of mobile and affordable embedded devices.

Keywords: *Convolutional neural networks, Traffic sign detection, RetinaNet, Deep learning.*

Introduction

Traffic Sign Detection has helped in increasing driver alertness and hence avoiding accidents, by using advanced driver assistance systems. Sometimes the traffic signs are not detected properly due to various challenges like illumination, rotation invariance, blurriness shape, and color change of traffic signs. The first sort of strategy for detecting traffic signs uses conventional image processing methods, whereas the second type relies on deep learning-based approaches. Traditional and deep learning-based approaches differ mostly in how well they perform as the size of the data rises. Because deep learning-based approaches require a significant quantity of data to fully grasp them, they perform poorly when our data is little. The constraints of traditional approaches are that we must extract features, that features are not generic, and

Corresponding Author:
Deepika
Research Scholar, Department of Multi-Disciplinary Engineering, The NorthCap University, Gurugram, India
email: deepika17ecd001@ncuindia.edu

that features must be extracted based on the domain of problem 2. Traffic sign detection plays an important role in real-world applications such as autonomous driver assistance systems (ADAS) [1,2] . The paper uses various meta architectures like faster RCNN, SSD, and Yolo for object detection. Further, these meta architectures are combined with various feature extractors like resnet-50,101, inception V2, and inception resnet V2 for better results. In the end, the author fine-tunes the already available model of Faster RCNN on the available dataset GTSDB. Mogelmose et al. present a review of traffic sign detection literature and introduce a publically available dataset of U S traffic signs. In current years the various state-of-art object detection algorithms like Yolo [3], SSD [4], R-FCN [5], and faster R-CNN [6], these authors utilize the convolutional neural networks and set into position in mobile devices.

Zhang et al. in their paper [7] predicted that some bad signs are detected due to the above challenges. Here the author also uses data augmentation techniques to justify the effectiveness of the algorithm and improve its accuracy. the author uses data augmentation only because they have a small volume of data available to us. the author divides the data into three parts basically for the development and tuning of parameters and the performance evaluation of the developed model by Ellahyani [8] presents a comparative analysis between color-based and shape-based methods for the intelligent transportation system. He also proved that both color-based and shape-based methods are not robust like in the case of the night the visibility of signs is not proper or color and shape also change. so due to all these reasons, people go with appearance-based methods.

The second approach to detect and classify traffic signs is to use deep neural networks. Maldonado [9] in their work uses the algorithm of support vector machines to classify traffic signs. Aghdam et al.[10] presents a traffic sign detection and categorization based on end-to-end CNN architecture. The author [11] utilizes the method of transfer learning author fine-tunes the models with the GTSDB dataset for the detection and classification of traffic signs based on their color and shape. Zang et al. [12] use the histogram of oriented gradients and also follow the sliding window approach for traffic sign detection. This method has one drawback it consumes a lot of time. Zhu et al. [13] make use of Convolutional neural networks for traffic sign detection to minimize the computational complexity of the algorithm and boost its speed. the spatial pyramid pooling networks in He, Kaiming, et al. [14] were also used to improve the efficiency of R-CNN by sharing computation. to reduce the computational complexity and increase the efficiency of the system many people prefer transfer learning techniques to achieve such goals.

The traffic signs are also detected based on color-based or shape-based features. traffic signs have their particular shape and color so that we can easily identify their sign from any particular image. Wang et al. [15] use the histogram of oriented gradients and the sliding window approach for traffic sign detection. The author proves that their results are robust to various conditions like bad lightning conditions, partial occlusion, and low-quality deformation. Deepika et al.2018 [16] prove that the features used are not suitable for the task of recognition. therefore, to overcome the above issues the author uses speed-up robust features(SURF) which are invariant to rotation, occlusion, and skew of the sign.

Two RetinaNet models are studied and contrasted in this article. To extract features from the pictures of traffic signs, these models used backbone CNN models called ImageNet pre-trained ResNet-50 and ResNet-101. The massive Tsinghua-Tenscent 100k dataset is used to train the RetinaNet models from scratch at first. After being trained, the models are further refined using data from the German Traffic Sign Detection Benchmark (GTSDB). The RetinaNet object detector using ResNet-50 and ResNet-101 achieved a mean average precision (mAP) of 94.28% and 95.70%, respectively, on the Tsinghua-Tenscent 100k dataset. In addition, the fine-tuned RetinaNet model based on transfer learning and employing ResNet-50, ResNet-101, and CNN obtained a competitive mAP of 96.47% and 97.49% on the GTSDB dataset, respectively. The outcomes show how well the suggested transfer-learning-based technique is in detecting traffic signs using the RetinaNet object detector memory. In summary, the major contributions of our work are summarized as follows:

- Evaluation of RetinaNet object detector using different ResNet backbones (ResNet-50, ResNet-101) for traffic sign detection.
- Use of deep learning transfer learning scheme for training the RetinaNet object detector on small sample size traffic sign detection (TSD) datasets.
- Numerous tests using three well-known TSD

benchmark datasets and comparisons with state-of-the-art techniques in terms of mean average precision (mAP), the number of model parameters, processing speed, and memory footprint.

The remaining sections are arranged as follows: RetinaNet and backbone CNN architectures are covered in detail in Section 2 along with ideas for methods. The performance assessment of the RetinaNet object detector using various backbones on the benchmark TSD dataset is shown in Section 3 along with the findings. Section 4 also examines the outcomes of comparing the suggested framework with modern, cutting-edge object detectors for TSD. Finally, Section 5 of the research provides some final reflections.

Proposed Methodology: The following segment defines the description of dataset and its configuration which is used in different Convolutional neural networks for resolving issues with traffic sign detection. Huang et al [17] explain that the experimental arrangement is composed of four meta-architectures (faster R-CNN, R-FCN, SSD, and YOLO) and six convolutional neural networks for feature extraction (Resnet v 150, Resnet v 101, inception v2, inception Resnet v2, mobile net v1, darknet – 19). Feature extractors use convolutional neural networks to extract premium features from images.

Due to a lack of time and computational expense, all the studies discussed in this paper use models that are publicly accessible and prepared for the benchmark data set [18] . With the help of the GTSDB benchmark data set, we fine-tune these models using the transfer learning methodology. This is achieved in such a manner that the effects of the identification and assessment are based on criteria of form and color, which can be further categorized as obligatory, prohibitive, and dangerous. When writing this article, both models are used that are pre-trained and are also available in the official repositories of the tensor flow object detection API.

Datasets: In countries like TK-101 and GTSDB, some of the benchmark traffic sign datasets have been gathered. The German Traffic Sign Detection Benchmark dataset (GTSDB) is the primary focus of our paper's analysis [19] . We prefer this dataset above others for many reasons, including the fact that it is commonly used in the survey to differentiate between various traffic signal identification techniques and is well-recognized. The fact that researchers from many fields contribute their results and evaluate the GTSDB Dataset is an additional challenge for the writers and the organization behind it. These days, our GTSDB benchmark dataset is kept where approval of results is preserved and ordered, and accessible on the leader table. Related grades allow them to explore state-of-the-art methodologies to identify traffic signals. And if their processing time is not counted. The GTSDB holds the Daily Traffic Scenes which are tape – records provided in various types of highways, such as highway, agricultural, urban, daytime, and half-light, and countless weather situations.

A collection of these categories of traffic signs are found in the benchmark datasets whose details have been discussed below.

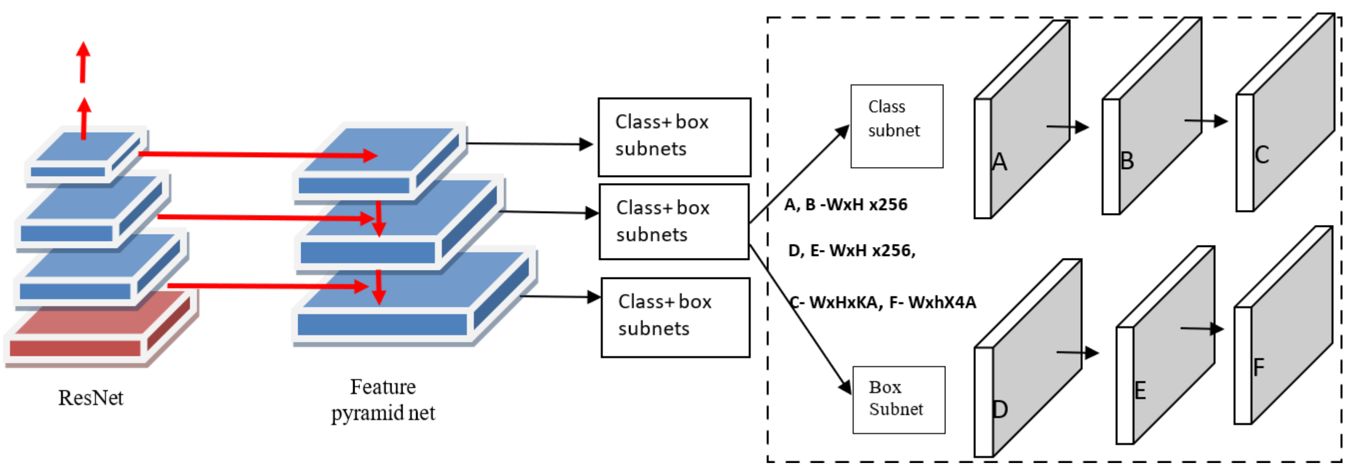

Figure 1: The Architecture: Resnet + Feature Pyramid Network + 2 Fully Connected Networks.

The German traffic sign detection benchmark (GTSDB) dataset: The German Traffic Sign Recognition Benchmark (GTSDB) dataset was made from 10 hours of video captured during daytime driving on various road types in Germany. The camera files used to create the traffic signal photos have a 1360 x 1024 pixel resolution. The video sequences go through a raw Bayer pattern processing procedure. 51,840 photos from 43 groups comprise the final database after annotation and gathering. The traffic signals have a resolution that varies from 15x15 to 222x193 pixels. To undertake to test phase, the entire data set was divided into a training set with 39209 photographs and a test set with 12630 shots.

The Tsinghua-Tenscent 100K dataset: There are a total of 45 classifications in the Tsinghua-Tenscent dataset, which consists of around 10,000 photographs with at least one traffic sign and 90,000 background images. These photos represent a wide range of lighting and weather situations. Every sign in the benchmark has a class label, bounding boxes, and pixel mask annotations.

RetinaNet object detector: RetinaNet is a one-stage object detector but it has the performance of a two-stage object detector. RetinaNet is a feature pyramid network with the cross-entropy loss replaced by focal loss. It has the best accuracy among single-stage and two-stage algorithms.it is faster than two-stage algorithms but still very slower than YOLO. the two-stage algorithms remove easily during the first stage. RetinaNet adds Focal Loss that discards easy background. The architecture of RetinaNet consists of three parts which are Resnet also with Feature Pyramid Network two Fully Connected Networks.

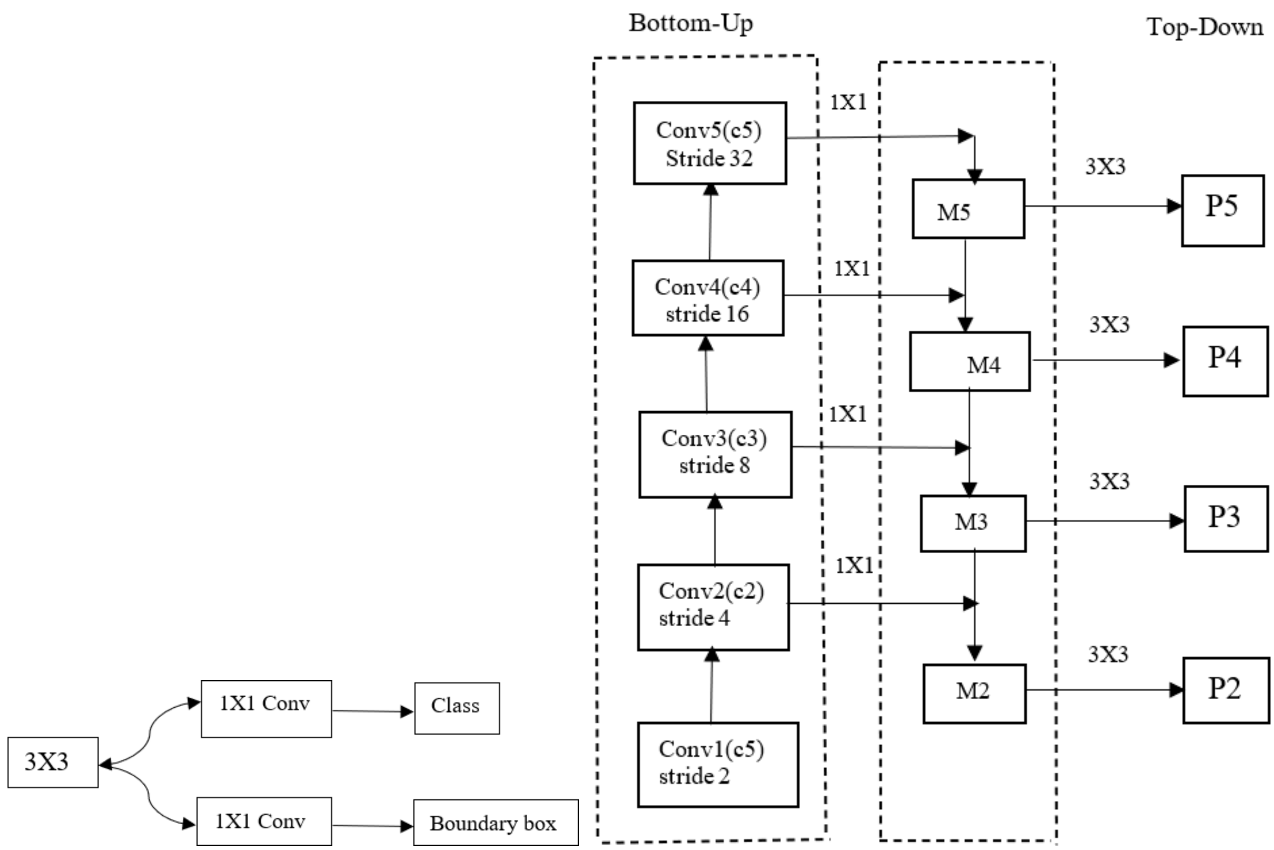

Figure 2: Feature Pyramid Networks.

Feature pyramid network: The function pyramid network is invariant in size and shares feature maps between multiple layers of the reset. The feature map is a mixture of fine-grained features and high-level features. The subnets exchange parameters at each level of Fully Connected Networks due to convolution only the input feature maps may be of various sizes. From a single input picture in resolution, Feature Pyramid Network (FPN) is used on top of Resnet to build a wealthy multi-scale feature pyramid network.

Focal loss: The Focal Failure is intended to fix the issues with the disparity of single-stage target detection where there are a very large number of potential context classes and only a few foreground classes. This makes training to be inefficient when most areas are negative and do not have a beneficial signal and the enormous amount of these negative examples overpower the training and decrease the output of the model. Focal loss is dependent on cross-entropy loss as seen below, and from well-classified cases, we can decrease the loss contribution by changing the gamma parameter. Zhu et al. [13] proposed traffic sign detection using focal loss for detecting traffic signs.

Focal Loss is designed to downweigh the loss. Let us take the example with two classes i.e., Foreground and Background. we are not able to train a single-stage object detector to be as accurate as two-stage detectors, so the contributions to this problem are given by RetinaNet and Focal Loss. previous results show that random resampling at 1:3 and hard negative resampling at a ratio of 1:3. both above solutions mean that in each step there are only samples that matter. instead of including all samples but using a different weight for each class. The regular class entropy is given by

$$CE(pt) = -log(pt) \tag{1}$$

Where the regular class entropy is given by $CE(pt) = -\alpha log(pt)$ and pt = $\begin{cases} p & if\, y = 1 \\ 1-p & otherwise \end{cases}$. The cross-entropy and the focal loss are given below as

$$CE(p, y) = \begin{cases} -log(p) & if\ y = 1 \\ -\log(1-p) & otherwise \end{cases} \tag{2}$$

so according to focal loss, every sample is weighted according to its error. We want to focus on samples that are mislabelled. In the above equation if y $\in \{\pm 1\}$ specifies the ground truth class and $p \in [0,1]$ is the model's estimated probability for class with label y=1. For national convenience we define pt: $\mathrm{y} = 1.\ \mathrm{pt}$:

$$Pt = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise \end{cases} \tag{3}$$

Results

Here, we provide a summary of the findings from the experiments on traffic sign detectors that are described in Section 3. For each experiment, accuracy, the number of parameters, the number of floating-point operations (FLOPs), memory use, and processing time are all assessed. On a machine with an Intel Core i7-4770 CPU, 16 GB of RAM, and an NVIDIA Titan Xp discrete GPU with 3840 CUDA cores and 12 GB of RAM, the models are trained and assessed. Here, we created tools using Tensorflow Object Detection API, Darknet 3, Darkflow 4, and Darkflow (Zang et al. 2016).

Table 1: Performance evaluation results on the gtsdb dataset (%)

Sign type	No. of instances	Precision (ResNet-50)	Precision (ResNet-101)
Prohibitory	161	99.81	99.76
Mandatory	49	94.64	96.12
Danger	63	94.91	96.60

Table 2: Comparison results of the proposed traffic sign detection scheme with the state-of-the-art methods on the GTSDB dataset.

Method	Year	Backbone	Frames Per Second (FPS)	Average Precision (AP)	GPU
Mask RCNN [14]	2020	ResNet-101	5	97.10	-
Cascade R-CNN [7]	2020	VGG-Net	-	96.80	NVIDIA GeForce GTX 1080 Ti GPU
Faster RCNN [6]	2017	VGG-16	10	96.10	-
Multi-Scale Cascadded RCNN [7]	2020	ResNet-50	12.50	96.45	-
Faster RCNN, [1]	2018	ResNet-50 ResNet-101	9.61 8.11	91.52 95.08	NVIDIA Titan XP Discrete GPU

Method	Year	Backbone	Frames Per Second (FPS)	Average Precision (AP)	GPU
Ours	proposed	ResNet-50	12.5	96.45	NVIDIA Titan XP Discrete GPU
Ours	proposed	ResNet-101	10.0	97.45	NVIDIA Titan XP Discrete GPU

On the GTSDB dataset, Table 2 displays the comparative findings of the most recent state-of-the-art techniques for detecting traffic signs. Among the existing methods, the proposed traffic sign detection framework using ResNet-101 attained the best performance with mean average precision (mAP) of 97.45%. On the GTSDB dataset, the Mask RCNN using ResNet-101 proposed by He et al. has attained an mAP of 97.10 with a processing speed of 5 FPS. Also, the cascade R-CNN model introduced by Zhang et al. has attained an mAP of 96.80% using the VGG-Net backbone. The proposed RetinaNet object detector using ResNet-50 and ResNet-101 backbone has attained an mAP of 96.45% and 97.45%, respectively. Figure 4 displays the outcomes of the RetinaNet model's prediction on the test photos chosen at random from the GTSDB dataset using ResNet-50 and ResNet-101backbone. The prediction results show the robustness of the proposed TSD framework using RetinaNet.

Detection results for GTSDB:

(a) RetinaNet Resnet-50

(b) RetinaNet Resnet-101

Figure 4: Examples of detections from two different models on a road with tiny, medium, and large traffic signs of varied sizes. All detections under these circumstances are precise. The third image from A, B, and C only depicts three traffic signals, yet one is still obscured.

Conclusions and Discussion

The report provides a summary of the present state of traffic sign detection. We primarily focus on the recognition of traffic signals rather than the entire TSD river. This article contrasts and compares eight deep neural network-based traffic sign detectors. Accuracy, speed, memory use, the amount of floating-point parameters, and the number of parameters that can be learned inside the CNN are some of the main characteristics that we examine. All of the models were pre-trained using the Microsoft COCO dataset, and they were then fine-tuned using the GTSDB dataset to help in the identification and categorization of traffic sign superclasses based on characteristics like shape and color: prohibitory, required, and hazardous. The lack of publicly accessible picture datasets that can be used to efficiently train and test algorithms is one of sign detection's drawbacks. In the current scenario, every new idea uses a new dataset for testing which makes comparisons between papers hard. The proposed traffic Sign Detection system reveals above-par performance under various challenging scenarios such as changes in illumination, scale variation, and rotation variation. Our system is capable of detecting traffic signs in acquired images irrespective of faded color and distorted shape.

The proposed system achieved a detection precision of 99.81% for the GTSDB Dataset. The system's soundness is validated by detecting the traffic signs under various challenging scenarios such as changes in illumination, scale variation, rotation variation, and similar color and shape variation.

Conflict of Interest: All authors declare that they have no conflict of interest.

Funding: No Funding.

Ethical Clearance: Not Applicable.

References

1. Arcos-García Á, Álvarez-García JA, Soria-Morillo LM. Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing. 2018 Nov 17;316:332-44.
2. Mogelmose A, Trivedi MM, Moeslund TB. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems. 2012 Oct 19;13(4):1484-97.
3. Redmon F. Redmon J., Farhadi A., Yolo9000: Better, faster, stronger. InProceedings of the IEEE conference on computer vision and pattern recognition (CVPR), IEEE Computer Society 2017 (pp. 6517-6525).
4. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox detector. InComputer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 2016 (pp. 21-37). Springer International Publishing.
5. Dai J, Li Y, He K, Sun J. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems. 2016;29.
6. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;28.
7. Zhang J, Xie Z, Sun J, Zou X, Wang J. A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. IEEE access. 2020 Feb 7;8:29742-54.
8. Ellahyani A, El Jaafari I, Charfi S. Traffic sign detection for intelligent transportation systems: a survey. InE3S web of conferences 2021 (Vol. 229, p. 01006). EDP Sciences.
9. Maldonado-Bascón S, Lafuente-Arroyo S, Gil-Jimenez P, Gómez-Moreno H, López-Ferreras F. Road-sign detection and recognition based on support vector machines. IEEE transactions on intelligent transportation systems. 2007 Jun 4;8(2):264-78.
10. Aghdam HH, Heravi EJ, Puig D. A practical approach for detection and classification of traffic signs using convolutional neural networks. Robotics and autonomous systems. 2016 Oct 1;84:97-112.
11. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?. Advances in neural information processing systems. 2014;27.
12. Zang D, Zhang J, Zhang D, Bao M, Cheng J, Tang K. Traffic sign detection based on cascaded convolutional neural networks. In2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD) 2016 May 30 (pp. 201-206). IEEE.
13. Zhu Y, Zhang C, Zhou D, Wang X, Bai X, Liu W. Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing. 2016 Nov 19;214:758-66.
14. He K, Zhang X, Ren S, Sun J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence. 2015 Jan 9;37(9):1904-16.
15. Wang G, Ren G, Wu Z, Zhao Y, Jiang L. A robust, coarse-to-fine traffic sign detection method. InThe 2013 international joint conference on neural networks (IJCNN) 2013 Aug 4 (pp. 1-5). IEEE.
16. Vashisth S, Saurav S. Histogram of Oriented

Gradients based reduced feature for traffic sign recognition. In2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2018 Sep 19 (pp. 2206-2212). IEEE.

17. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 7310-7311).

18. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: Common objects in context. InComputer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 2014 (pp. 740-755). Springer International Publishing.

19. Stallkamp J, Schlipsing M, Salmen J, Igel C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. Neural networks. 2012 Aug 1;32:323-32.