Faster R-CNN based Traffic Sign Detection and Classification

MONIRA ISLAM, MD. SALAH UDDDIN YUSUF
Department of Electrical and Electronic Engineering
Khulna University of Engineering & Technology
Khulna-9203, Khulna
BANGLADESH

Abstract: - Traffic sign is the key aspect in road and also for the autonomous car. Detection and classification of these sign plays a vital role for the invention of driverless vehicles. Convolutional neural network (CNN) has the ability to learn local features using series of convolutional and pooling layer observing the image sequences. In this work, traffic sign detection and classification has been performed based on deep learning approach. The experiment conducted on Germen Traffic Sign Detection Benchmark (GTSDB) and Recognition Benchmark (GTSRB) for detection and recognition. For traffic sign detection a two-stage detector, Faster R-CNN with ResNet 50 backbone structure is used where the CNN layers extracted the features of traffic signs from the images and the region proposal network (RPN) filter the object from the image to create bounding box based on the extracted feature map. The classification network classifies the traffic signs and predict the proposal confidence score. A general deep learning model is transferred into a specific output with weights with transfer learning by tuning the pretrained model based on COCO image dataset. The performance is compared with ResNet 152, MobileNet v3 and RetinaNet based on the confidence score and mean average precision (mAP). Faster R-CNN with ResNet-50 shows better detection performance comparing with other backbone structure. In addition, a series of convolution layer with batch normalization followed by max pooling layer is used to build a classifier and softmax is used in the output for 43 class classification and 97.89% test accuracy has been obtained.

Key-Words: - Traffic sign detection, faster R-CNN, ResNet 50 FPN, MobileNet v3 large FPN, traffic sign classification, residual network

Received: February 17, 2021. Revised: December 13, 2021. Accepted: January 20, 2022. Published: February 10, 2022.

1 Introduction

As technology advances, traffic sign detection and recognition become very important in case of automatic driving system, and it is an active research topic in real world scenario. Automatic detection of traffic signs is essential for autonomous vehicles, driverless car, or robotic car. Additionally, for general traffic system it may reduce driver's effort as well as mitigate the risk of accident at foggy weather or adverse environmental condition. To build a traffic sign recognition system, traffic sign detection and recognition are the two important stages. Traffic sign detection and classification from the captured image is a real world problem that has been dealt with in this paper. For many years, the local feature descriptor dominates all the research area of computer vision. A series of traditional classifiers were used for object detection that were trained with various manually designed features obtained from histogram of oriented gradient (HOG) or HAAR wavelet features [1]. HOG and HAAR features with support vector machine (SVM) are the traditional approach to deal with recognition problems in terms of computer vision application. SVM classifier was used to perform 3D object detection on the image regarding as points of a high dimensional space in [2]. These features lack in diversity and may change with time which limits the performance in terms of accuracy due to motion blur, light, or other environmental factors. With the advancement of deep learning technology, object detection has become active field of research that aims end-to-end optimization and requires more computational power and cost. Convolution neural network (CNN) based object detection methods achieved better performance.

Currently, research on traffic sign classification has drawn much attention in autonomous vehicles. In literature, different deep learning methods were applied for this task. The deep learning networks extract the high-level features that can achieve better performance in terms of detection and classification. CNN-SVM based method is proposed to classify traffic signs where the convolution layer extracts the temporal features and SVM classifies the signs based on the deep learning features [3]. In [1], a new method was proposed using HOG-SURF features and CNN was used for classification. The HOG features are global descriptors and SURF are rotational invariant features. These hybrid feature built higher dimensional feature vectors and the convolution layers effectively extracted deep learning features yielding 98.48% accuracy. A deep learning approach was adopted for mobile robotics detection where scale invariant features (SIFT) and SURF features were used to reduce the feature dimension as PCA algorithm [4]. They applied CNN and attained great advantage over traditional machine learning model. Before the adoption of CNN, various methods were adapted for traffic-sign classification, e.g., based on SVM [5] and sparse representations [6]. To detect traffic signs at different environmental settings and blur images, different complicated algorithms were addressed which required long training time. Multi-scale deconvolution network (MDN) with multi-scale convolutional neural network with deconvolution sub-network was proposed in [7] that efficiently localized the features for detecting traffic signs and enhanced reliability to train the model. In [8], a fast and precise traffic sign recognition system was proposed based on two CNN networks using Swedish traffic sign database. One of that was for region proposals of traffic signs and other for classification of each region. In [9], low-cost embedded system was developed using Raspberry Pi module based on HOG features to detect traffic signs. Authors in [10] used different pre-trained CNN model to show the effectiveness of deep learning model for medical image classification. Then they applied fine-tuning to evaluate the importance of transfer learning for this task. To detect objects in real world scenario, CNN based two stage object detection methods i.e., Region-based Convolutional Neural Network (R-CNN), Fast R-CNN [11], Faster R-CNN [12] were proposed that used the feature maps to build region proposal first and then detected the object by locating the bounding regression box. In [13], proposal free SSD model was used based on VGG-16 architecture. YOLO is also popular single stage object detector that was used in [14]. R-CNN shows better accuracy but limits the performance for the slow processing speed. Faster R-CNN algorithm is effective over R-CNN and Fast R-CNN in terms of faster training and speed up the detection performance [15].

CNN outperforms other classifiers for detecting objects while tested on the GTSRB benchmark. A committee of neural network [16], multi-scale CNNs [17] with hinge loss [18] was adopted for this task that achieved better performance than handcrafted feature-based approach in [19]. It is observed that CNN are inherently efficient when used in a sliding window fashion, as many computations can be reused in overlapping regions [20]-[21]. CNN was demonstrated to determine an object's bounding box together with its class label. Another widely used strategy for object detection using CNNs is to first calculate some generic object proposals and perform classification only on these candidates. Selective search takes about 3s to generate 1000 proposals for the Pascal VOC 2007 images [22] whereas the more efficient Edge Boxes approach still takes about 0.3 s [23]. In addition, it applies a deep CNN to every candidate proposal, which is very inefficient and time-consuming. In this paper, traffic signs are detected based on the two-stage detector, Faster R-CNN object detection algorithm using GTSDB database that overcome the limitations of R-CNN and Fast R-CNN. This two-stage detector show their effectiveness over one-shot detector in terms of mean average precision (mAP) and losses. Faster R-CNN with ResNet-50 Feature Pyramid Network structure depends on the feature activation output of each stage's last residual block. The focus of this study is to track and recognize the traffic signs from the image. Then, 2D CNN architecture is used to recognize the traffic signs from GTSRB database to classify 43 classes of traffic signs.

The rest of the paper is organized as follows. Proposed methodology is described in section II and section III shows the experimental results with analysis. Finally, conclusion is stated in section IV.

2 Proposed Methodology

2.1 Traffic Sign Detection and Classification Dataset

In this work, traffic sign has been localized and classified using object detection techniques based on deep learning. For this purpose, Germen Traffic Sign Detection Benchmark (GTSDB) and Germen Traffic Sign Recognition Benchmark (GTSRB) two freely available dataset have been used. GTSDB is used to detect the traffic sign and there are 900 images consisting of street objects and traffic signs and the non-target images are eliminated for further processing. Faster R-CNN is used to detect the traffic sign with regression box analysis and classify them with the classifier. The other dataset GTSRB is used containing 43997 images of traffic signs. These images are categorized into 43 classes.

2.2 Faster R-CNN Algorithm for Object Detection

The R-CNN algorithms are two stage object detection methods that are all region based. In the first stage the model proposes a set of regions of interests (ROI) with the region proposal network (RPN) and in the second stage a classifier processes the region candidates to classify the targets. In R-CNN, category independent region proposals are obtained via selective search. For each image region, one forward propagation through CNN generates a feature vector. The selective search algorithm is a fixed algorithm that is unable to learn the features which might generate the bad candidate region of proposals. Additionally, the localization of candidate box might be inaccurate for detecting the objects. Fast R-CNN algorithm resolved some issues related to R-CNN by processing the input image and enhancing the processing speed. However, ROI is generated by selective search and the region proposals are not learnt from the training data for Fast R-CNN. Faster R-CNN algorithm overcome the limitations of Fast R-CNN by designing the RPN that generates the region proposals after training. After RPN training, the ROI pools features from the proposals for training Fast R-CNN model with a classification head and a regression head. RPN provides different sized feature maps and ROI pooling reduce the feature map by generating the same size feature map and splits into a fixed number (k) or in equal number and apply max-pooling on every region. So, the output of ROI pooling is a fixed number regardless the size of input. This algorithm shares parameters for both the RPN and Fast R-CNN model and trains jointly. The training speed enhances with Faster R-CNN because time cost is less for generating region proposals with RPN than selective search. The RPN ranks the region boxes which are called anchor boxes and proposes the most likely ones that contain objects. The anchor boxes predict the anchors to be background and foreground and refines the anchors accordingly.

For Faster R-CNN, anchors play a vital role at a position of an image and work well for both Pascal VOC and COCO dataset. The ground truth box labels the anchors according to the overlapping region. The higher Intersection over Union (IoU) is termed as foreground or positive class whereas the lower IoU is termed as background or negative class. For each anchor there exists two possible labels called logit (foreground and background) which is feed into softmax to predict the labels. The training dataset is processed with feature map and labels in this way for training a classifier. The receptive fields of every positions on the feature map needs to cover all anchors to provide sufficient information for prediction. Figure 1 shows the basic architectural framework for Faster R-CNN algorithm.

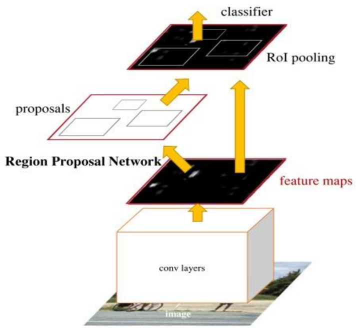


Fig.1 The framework of Faster R-CNN architecture for object detection [4].

2.3 Feature Pyramid Network (FPN)

Faster R-CNN ROI pools features from only the topmost feature maps. Their spatial size might be too small for localizing small-scale objects. To detect object, ROIs of different scales are needed to be assigned to the pyramid levels. Feature pyramid is a series of features with different scales, so that large and small objects can be detected from low-resolution and high-resolution feature maps respectively. FPN up-samples feature maps with bi-linear interpolation and add skip connections to additively fuse encoder-decoder feature maps. The feature maps are selected as following based on the width w and height h of the ground truth box,

$$k = k_0 + \log_2\left(\sqrt{wh}/hx\right) \qquad (1)$$

where, k_0 denotes the scale of topmost feature maps and k denotes the P_k layer of FPN to generate the feature patch.

2.4 Loss Function

For training RPN, training objective need to determine, and positive and negative label need to determine. The anchors are assigned as positive label if any of the two condition is satisfied. (i) The anchor that has higher IoU overlap with the ground truth box; (ii) The anchor that has IoU overlap higher than 0.7 is termed as positive label. If IoU ratio of any anchor is lower than 0.3 for all ground truth box, it is assigned as negative label. The anchor that is not labelled as positive or negative, do not contribute to training. The loss of the classifier comprises the classification and regression loss. Equation 2 represents the loss function for Faster R-CNN algorithm.

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*) \qquad (2)$$

where, i is the index of anchor in a mini-batch. p_i is the predicted probability of an anchor for being a target object and p_i^* is the ground truth label. t_i is a vector that represents 4 parameterized co-ordinates of predicted bounding box and t_i^* is the ground-truth bounding box. L_{cls} is the log loss classification loss and L_{reg}, t_i, t_i^* is the regression loss activated only when the anchor contains an object (i.e., ground-truth $p_i^* = 1$). t_i is the target of regression layer consisting of 4 variables t_x, t_y, t_h, t_w. Here, (x, y) is the coordinate of the bounding box center and h and w is the height and width of the box, respectively. The target of the regression box t_i^* can be obtained by parameterization of the four coordinates.

$$t_x = \frac{(x - x_a)}{w_a}, \quad t_y = \frac{(y - y_a)}{h_a}, \quad t_w = \log\left(\frac{w}{w_a}\right),$$

$$t_h = \log\left(\frac{h}{h_a}\right) \quad t_x^* = \frac{(x^* - x_a)}{w_a}, \quad t_y^* = \frac{(y^* - y_a)}{h_a},$$

$$t_w^* = \log\left(\frac{w^*}{w_a}\right), \qquad t_h^* = \log\left(\frac{h^*}{h_a}\right) \qquad (3)$$

2.5 Faster R-CNN with ResNet FPN for Object Detection

For object detection, deeper network like ResNet 50, ResNet 152 bottleneck architecture will be used that contains 4 stages and performs the initial convolution and max- pooling using 7×7 and 3×3 kernel sizes respectively. For each stage, the network contains 3 residual blocks having 3 layers stacked one over other. The 3 layers are 1×1, 3×3, 1×1 convolution. The FPN extract the feature maps and feed to the RPN. In FPN, for each scale level, 3×3 convolution filter is applied over the feature maps followed by 1×1 convolution for object prediction and boundary box regression called RPN head.

The 1×1 convolution layer is responsible for reducing and then restoring the dimensions and the 3×3 layer is left as a bottleneck with smaller input/output dimensions. With the progress of each layer the input will be reduced to half whereas channel width will be doubled. Finally, the network has an average pooling layer followed by a fully connected layer for object detection. Figure 2 shows the basic structure of FPN with RPN implemented on ResNet architecture.

2.6 Traffic Sign Detection with Faster R-CNN

In this work, traffic sign is detected from the image. Faster R-CNN performs better for PASCAL VOC dataset for detecting 20 objects. But detecting small objects like traffic sign from images even on different environmental condition like motion or blur. So, the parameters need to adjust to achieve for better performance. 43 classes of traffic signs were in GTSDB. The model was pretrained with COCO dataset to initialize the parameters of RPN network and Fast R-CNN. Faster R-CNN is trained by GTSDB dataset which have fewer levels than COCO dataset. When the number of datasets is limited then transfer learning is required [24]. The RPN network generates the region proposals and non-maximum suppression strategy is applied to score the anchors and passes the positive and negative labels to RoI pooling layer. After training, the loss of classifier and regressor is obtained and the confidence score shows the object detection performance in terms of testing image according to the amount of input.

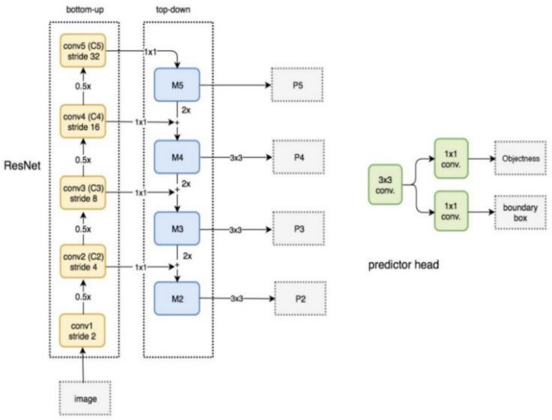


Fig.2 FPN with RPN based on ResNet architecture

2.7 CNN for classification

A 2D CNN model is built for classification. CNN is better due to the convolution of 2D kernel that effectively extracts the discriminatory temporal features for classifying the objects [25]. In this work, three Convolution layer with Relu activation function, followed by Batch normalization, max-pooling and dropout layer are used for building the CNN model for recognition of traffic sign using GTSRB dataset. The batch size is 32 with learning rate 0.0001. Back propagation algorithm is used for effective training that adjust the parameters minimizing the cost function. These updated parameters can be obtained from chain rule through the gradient of each parameter with activation function and input vector in each neuron. The 2D convolution extracts the features in time domain that classify the traffic signs efficiently.

3 Analysis of Experimental Result

In this work, traffic signs are detected trained on Faster R-CNN model based on GTSDB. The non-target images are rejected, and 506 images are used for analysis where the images size are 1380 × 800 pixels. 406 images are used for training and 100 for testing. The experiment is performed on Google Colab dedicated GPU platform. The Resnet 50 architecture is used to detect the object and the performance is compared with MobileNet or RetinaNet or shallow ResNet 152 architecture. mAP and the loss of classifier, regressor are observed as the performance indicator of each method.

3.1 Performance with Faster R-CNN based on Resnet FPN Architecture

The experiment is performed to detect traffic sign detection using GTRDS database with Faster R-CNN which overcome the limitations of R-CNN and Fast R-CNN. The ResNet 50 FPN and ResNet 152 FPN shallow layers are implemented as network structure. Stochastic gradient descent (SGD) is used as optimizer with weight decay and 0.0005 initial learning rate. Cosine annealing warm restart learning scheduler is used. As number of iterations increase the learning rate are reduced. The training and testing time per iteration are 0.754 sec and 0.157 sec respectively whereas required time for accumulating evaluation result is 0.06 sec.

The results of Faster R-CNN with backbone Resnet architecture are shown as bounding box to capture the traffic sign and the confidence score. The bounding box (bbox) is attained from RPN which allows to select the appropriate ROI and the confidence score measures the fitness of the box. The bbox indicates the localization of the predicted box with a class label. Figure 3 depicts the distribution of number of traffic signs with the class number. Figure 4 and Figure 5 show the confidence score of detected traffic signs in images with ResNet 50 and 152 network architecture respectively. Figure 4 shows that Faster R-CNN with ResNet 50 can detect the traffic signs effectively at foggy street, different lighting state or normal condition attaining equal or greater than 0.96 score at all conditions even with multiple signs at single frame. Figure 4(b) shows that

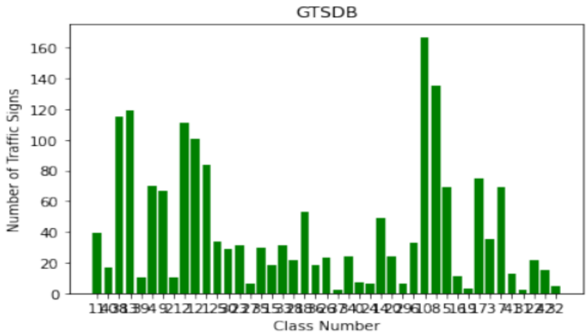
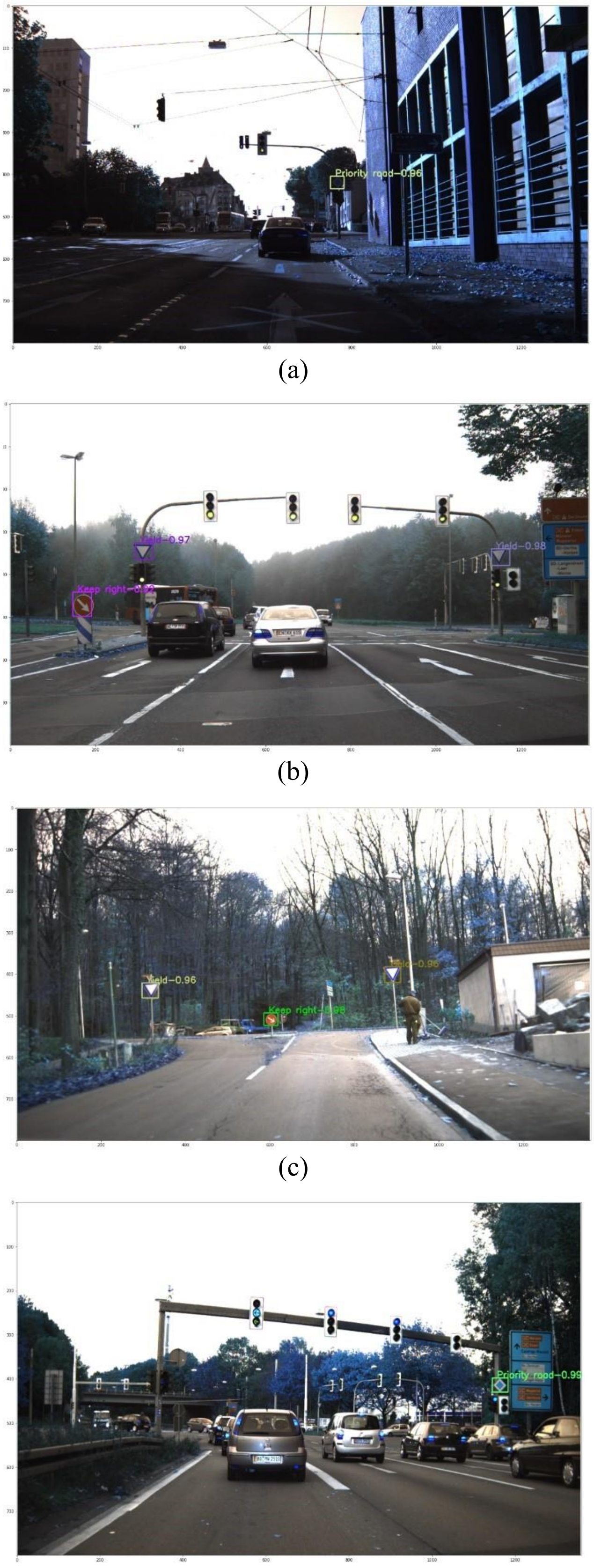


Fig. 3 Distribution of number of traffic sign with class number of GTSDB dataset



(a)

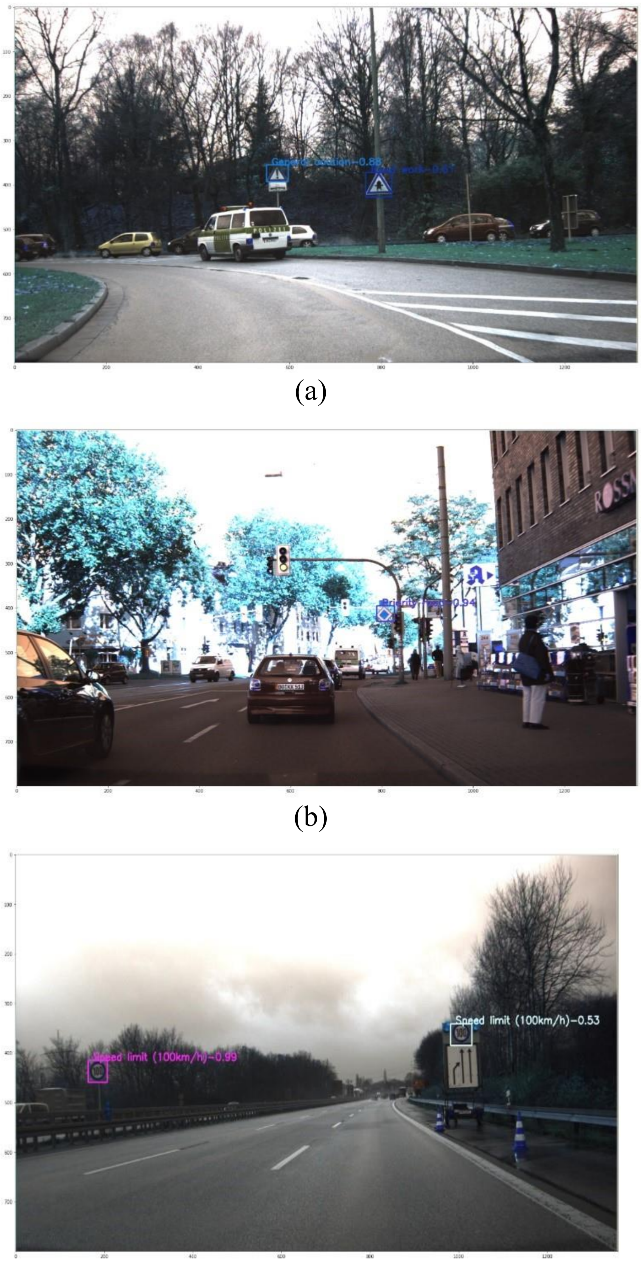
(b)

(c)

(d)

Fig. 4 Traffic Sign detection with Faster R-CNN based ResNet 50 FPN architecture, at (a) foggy street; (b) different lighting condition; (c) multiple traffic sign detection; (d) normal lighting condition.

at different lighting condition, the detector detects the three traffic signs with 0.97, 0.98 and 0.98 confidence score with accurate class label termed as "yield" on both left and right side of road and "keep right" direction respectively. The accuracy of localization falls with less confidence score when Faster R-CNN was used with Resnet 152 architecture. While there exist multiple targets in one image frame, score of one target is found 0.53 at foggy weather as shown in Figure 5(c) which is much lower than ResNet 50 FPN. Resnet50 backbone is found more suitable to detect the traffic signs at any environmental conditions. The loss of the regressor and classifier of Faster R-CNN with Resnet50 and 152 are shown in Figure 6 and Figure 7 respectively where the horizontal axis indicates the number of training epochs and vertical axis indicates the value of losses. It is observed the neural network trained for 35 epochs and converged with progression of iteration.

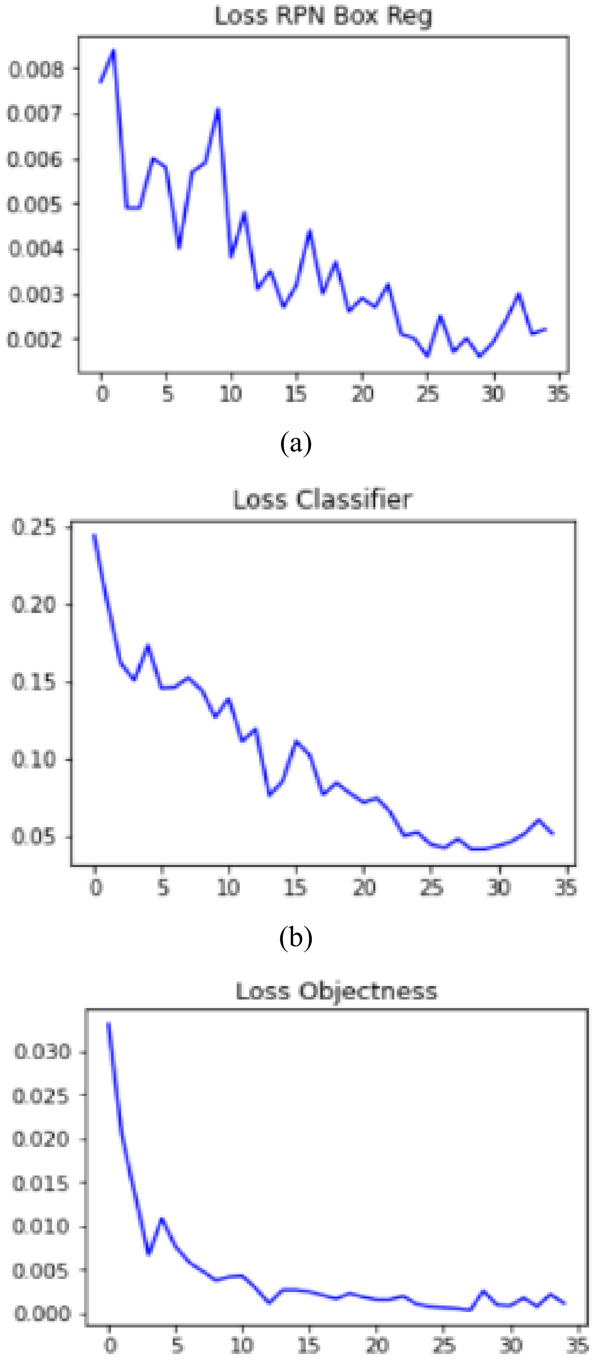


(a)

(b)

(c)

Fig.5 Traffic Sign detection with Faster R-CNN based ResNet 152 FPN architecture, at (a) different lighting condition; (b) normal lighting condition with multiple signs; (c) foggy weather.

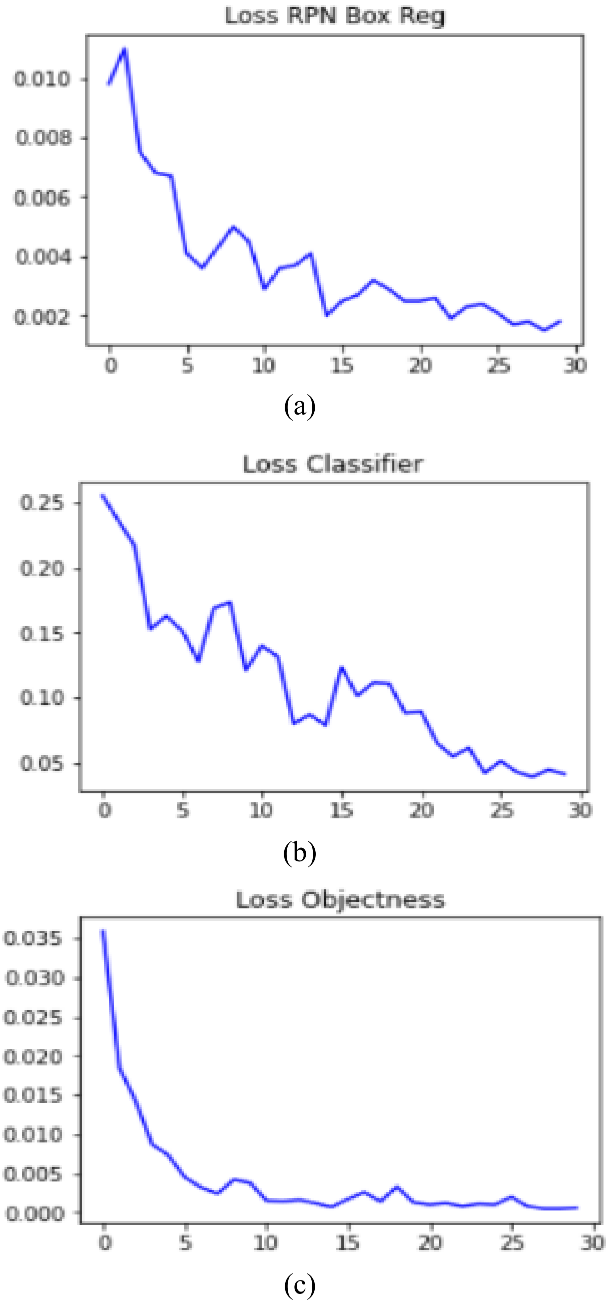


(a)

(b)

(c)

Fig.6 Losses of (a) RPN box regression; (b) Classifier; (c) objectness with ResNet 50 FPN architecture.



(a)

(b)

(c)

Fig.7 Losses of (a) RPN box regression; (b) Classifier; (c) objectness with ResNet 152 FPN architecture.

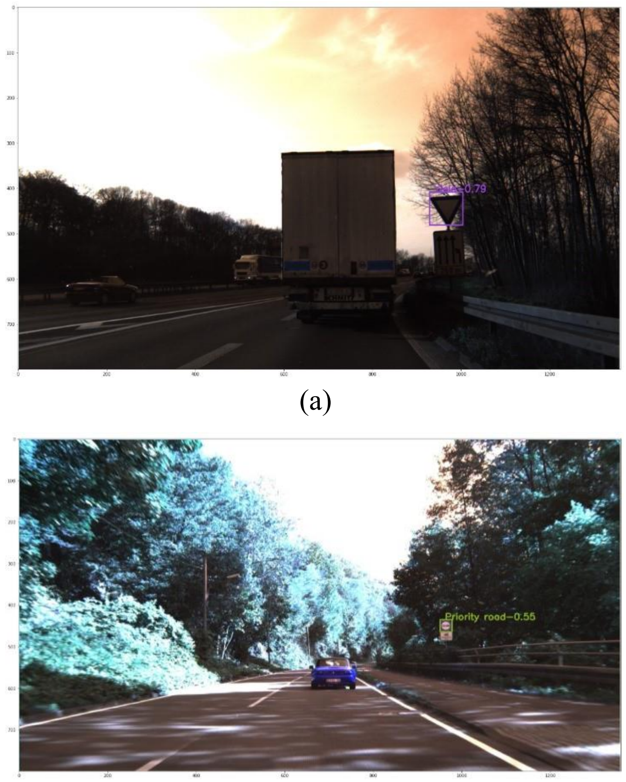
3.2 Performance with MobileNet and RetinaNet FPN architecture

The performance of traffic sign detection is compared with Faster R-CNN based MobileNetV3 Large FPN architecture and one-shot detector RetinaNet ResNet 50 FPN architecture is tabulated in Table 1. The losses of classifier, regressor as well as total losses are illustrated in Table 1. The loss is found small for a two-shot detector, Faster R-CNN whereas it increases notably for the one-shot detector, RetinaNet for both classification and regression while both detectors considered the Resnet50 FPN.

Table 1 Performance comparison of different network architecture in terms of losses.

Method	**Losses**	**$L_{classifier}$**	**$L_{regressor}$**
Faster R-CNN + ResNet 50 FPN	0.09	0.05	0.04
Faster R-CNN + ResNet 152 FPN	0.11	0.07	0.04
Faster R-CNN + MobileNet v3 large FPN	0.181	0.085	0.096
RetinaNet + ResNet 50 FPN	0.56	0.45	0.09

The losses of classifier and regressor are found 0.05 and 0.04 respectively for Faster R-CNN with ResNet 50 FPN which are the least among all the network architectures. Figure 8(a) and Figure 8(b) indicate the confidence score of detected traffic sign with Faster R-CNN+MobileNetV3 FPN and RetinaNet +ResNet FPN structure. It is observed that Faster R-CNN with lager FPN MobileNetV3 detects the object with confidence score 0.79 as shown in Figure 8(a). The performance declines with 0.55 score for RetinaNet with ResNet 50 backbone as displayed in Figure 8(b) which indicate the efficacy of two-shot detector, Faster R-CNN for the intended task than the one-shot detector RetinaNet.



(a)

(b)

Fig.8 Traffic sign detection with (a) MobileNet v3 FPN; (b) RetinaNet ResNet 50 FPN architecture.

Table 2 and Table 3 list the performance of different architecture in terms of mean average precision (mAP) and mean average recall (mAR) evaluation index for traffic sign detection. Precision measures the correctness of prediction and recall indicates the goodness of predicting the positives. IoU measures overlapping between the predicted boundary and the ground truth one whereby threshold is predefined as 0.5. Precision and Recall can be obtained as

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Average precision (AP) indicates the area under the precision-recall curve whereas the evaluation metric, precision or recall ranges in 0 to 1. In case of COCO dataset AP and mAP considers the same meaning to measure the accuracy of the object detector. Table 2 indicates that, the mAP for Faster R-CNN with ResNet 50 FPN is greater than other networks yielding 0.29, 0.34, and 0.64 for the small, medium, and large area respectively. The RetinaNet with ResNet 50 FPN provides larger mAR as 0.49 and 0.77 for small and large area respectively as listed in Table 3. Faster R-CNN with ResNet 50 architecture also attained a notable mAR for small, medium, and large area. The score of bbox, classification and regression losses, mAP and mAR evaluation metric show a clear advancement in the performance for Faster R-CNN with ResNet50 FPN network architecture than Faster R-CNN based other architecture. Additionally, the two-shot detector, Faster R-CNN outperforms one-shot detector, RetinaNet in this task for traffic signal detection and recognition.

Table 2 Performance comparison of different network architecture in terms of mAP for IoU 0.5:0.95.

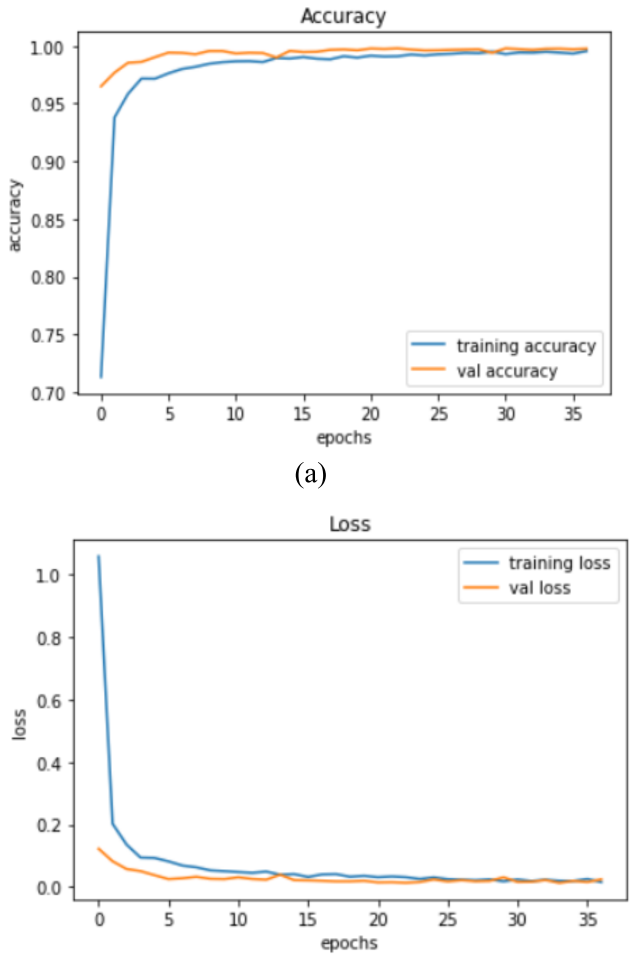
Method	**mAP Area$_s$**	**mAP Area$_m$**	**mAP Area$_l$**
Faster R-CNN + ResNet 50 FPN	0.29	0.34	0.64
Faster R-CNN + ResNet 152 FPN	0.16	0.28	0.46
Faster R-CNN + MobileNetV3 large FPN	0.085	0.15	0.33
RetinaNet + ResNet 50 FPN	0.087	0.19	0.17

Table 3 Performance comparison of different network architecture in terms of mAR for IoU 0.5:0.95.

Method	***mAR Area$_s$***	***mAR Area$_m$***	***mAR Area$_l$***
Faster R-CNN + ResNet 50 FPN	0.42	0.59	0.64
Faster R-CNN + ResNet 152 FPN	0.33	0.53	0.64
Faster R-CNN + MobileNetV3 large FPN	0.15	0.32	0.35
RetinaNet + ResNet 50 FPN	0.49	0.56	0.77

3.3 Traffic sign recognition with CNN

Figure 9 (a) and Figure 9(b) depict the accuracy and loss curve respectively of the CNN classifier for 35 epochs. The classification of traffic signs is performed with GTSRB dataset. A Classifier is built



(a)

(b)

Fig. 9 (a) Plot of training and validation (a) accuracy; (b) loss of the CNN classifier.

with convolution layer followed by batch normalization, maxpooling and dropout layer is used to classify the signs. The total data samples are partitioned randomly where 70% of the total samples were taken for training, 15% for testing and 15% for validation. The training, testing and validation accuracy is found 99.56%. 97.89% and 99.76% respectively with validation loss 2.48%. The precision, recall and F-score is 97.96%, 97.89% and 97.87% respectively on test data.

4 Conclusion

Traffic sign detection from street image is challenging because of motion, blur, object size or different lighting condition. Faster R-CNN based two stage object detection method has been applied based on different deep learning network structure. Faster R-CNN based ResNet 50 FPN network shows the improved result comparing with the other structure in terms of confidence score and mAP. The losses are higher for RetinaNet which is the one shot detector. The two-stage detector Faster R-CNN performs better for traffic sign detection despite motion, blur, fog or lighting condition than the one-shot detector which can be further implemented in real time scenario.

References:

[1] R. Madan, D. Agrawal, S. Kowshik, H. Maheshwari, S. Agarwal, and D. Chakravarty, Traffic Sign Classification using Hybrid HOG-SURF Features and Convolutional Neural Networks. In ICPRAM, pp. 613-620, 2019.

[2] M. Pontil, and A. Verri, Support vector machines for 3D object recognition. IEEE transactions on pattern analysis and machine intelligence, 20(6), pp. 637-646, 1998.

[3] Y. Lai, N. Wang, Y. Yang, and L. Lin, Traffic signs recognition and classification based on deep feature learning. In 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), Madeira, Portugal, 2018, pp. 622-629.

[4] A. Lee, Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics. Swarthmore College, 2015.

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

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

[7] S. Pei, F. Tang, Y. Ji, J. Fan, and Z. Ning, Localized Traffic Sign Detection with Multi-scale Deconvolution Networks, 2018.

[8] Y. Yang, S. Liu, W. Ma, Q. Wang, and Z. Liu, Efficient Traffic-Sign Recognition with Scale-aware CNN, 2018. arXiv preprint arXiv:1805.12289.

[9] A. Soetedjo, and I. K. Somawirata, An Efficient Algorithm for Implementing Traffic Sign Detection on Low Cost Embedded System. International Journal of Innovative Computing Information and Control, 14(1), pp. 1-14, 2018.

[10] S. Hoo-Chang, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, and R. M. Summers, Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. IEEE transactions on medical imaging, 35(5), pp. 1285, 2016.

[11] R. Girshick, Fast R-CNN, IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.

[12] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, Advances in neural information processing systems, 2015, pp. 91-99.

[13] W. Liu, D. Anguelov, D. Erhan, S. Christian, R. Scott, F. Cheng-Yang, B. Alexander, SSD: Single Shot MultiBox Detector, European Conference on Computer Vision, pp. 21-37.

[14] J. Redmon, S. Divvala, R. Girshick R, et al, You only look once: Unified, real-time object detection, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779-788.

[15] W. Cai, J. Li, Z. Xie, T. Zhao and K. LU, “Street Object Detection Based on Faster R-CNN,” Chinese Control Conference (CCC), 2018, pp. 9500-9503.

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

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

[18] J. Jin, K. Fu, and C. Zhang. Traffic sign recognition with hinge loss trained

convolutional neural networks. IEEE Transactions on Intelligent Transportation Systems, 15(5):1991–2000, 2014.

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

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

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

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

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

[24] M. Milicevic, K. R. U. N. O. S. L. A. V. Zubrinic, I. N. E. S. Obradovic, and T. O. M. O. Sjekavica, Data augmentation and transfer learning for limited dataset ship classification. WSEAS Trans. Syst. Control, vol. 13, no. 1, pp. 460-465, 2018.

[25] A. El-Sawy, M. Loey, and H. El-Bakry, Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, vol. 5, 2017, pp. 11-19.

Author Contributions:

Monira Islam: Conceptualization, methodology, simulation and the optimization, original drafting.

Md. Salah Uddin Yusuf: Simulation, formatting, technical review, editing and methodology.

All authors have read and agreed to the published version of the manuscript.

Sources of funding for research presented in a scientific article or scientific article itself

This research did not receive any grant from funding agencies in the public, commercial, or non-profit sectors.

Authors Biography

Monira Islam received her B.Sc. and M.Sc. degree from Khulna University of Engineering Technology (KUET), Khulna, Bangladesh and serving as an Assistant Professor at the same university. Currently, she is doing her PhD at The Chinese University of Hong Kong. Her research interest includes Brain-computer interaction, Bio-Signal processing, Image Processing and Computer vision.

Md. Salah Uddin Yusuf received the Ph.D. degree in Electrical and Electronic Engineering from Khulna University of Engineering and Technology (KUET), Bangladesh, in 2016. Since 2001, he has been with the same department of same University as a faculty member and currently serving as Professor. In professional activities, he is life fellow of Institute of Engineers (IEB), Bangladesh and member of IEEE. His research interests mainly focus on deep learning based signal, image and video processing with quality assessment, Face liveness detection and authentication, GAN-based Human activity analysis for modern healthcare system.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)