A Benchmark for Detection and Recognition of Bangladeshi Traffic Signs in Real-world Images

Authors

Rizwanul Haque Khan (170042078)

Md Saimul Haque Shanto (170042080)

Ahmed Nusayer Ashik (170042086)

Supervisor

Dr. Md. Hasanul Kabir

Professor, Department of CSE

Co-supervisor

Sabbir Ahmed

Lecturer, Department of CSE

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Software Engineering

Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Gazipur, Bangladesh

May, 2022

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Rizwanul Haque Khan, Md Saimul Haque Shanto and Ahmed Nusayer Ashik under the supervision of Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh and Sabbbir Ahmed, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Rizwan	Shanto	
Rizwanul Haque Khan	Md Saimul Haque Shanto	Ahmed Nusayer Ashik
Student ID: 170042078	Student ID: 170042080	Student ID: 170042086

Supervisor:

Dr. Md. Hasanul Kabir
Professor, Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Co-supervisor:

Sabbir Ahmed
Lecturer, Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Acknowledgement

We would like to start by expressing our deepest gratitude to Almighty Allah for allowing us to complete our research successfully. We are indeed grateful to be able to submit our thesis work by which are eventually going to end our Bachelor of Science study.

Next we would like to express our honest appreciation to Dr. Md. Hasanul Kabir, Professor, Department of Computer Science and Engineering, Islamic University of Technology and Sabbir Ahmed, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology for being our adviser and mentor. Their inspirations, ideas and observations for this particular research work were invaluable, and this thesis would have never been totally successful without their encouragement and proper guidance. Their valuable opinion, time and input provided throughout the thesis work, from the first phase of thesis topics introduction, research area selection, proposition of algorithm, modification and implementation helped us to do our thesis work in proper way. We are grateful to them for their constant and energetic guidance and valuable advice.

We extend our appreciation to all the respected jury members of the thesis committee for their thoughtful comments and constructive criticism of our research work. They certainly helped us develop this work of science. Finally, we would like to express our sincere gratitude to all the faculty members of the Computer Science and Engineering department of Islamic University of Technology for providing us with a congenial and supportive work environment.

Abstract

Traffic sign detection is an indispensable part of autonomous driving and transportation safety systems. However, the accurate detection and recognition of traffic signs remain challenging, especially under extreme conditions, such as various weather and geo-social features. Though a lot of work has been done in the domain of Traffic Sign Detection and Recognition (TSDR) systems, only a few of them focus on a dataset that comprises the real-world challenges. Moreover, in the context of Bangladeshi traffic sign detection, the research is in very preliminary stage and there is no publicly available dataset till date. The geo-social features of Bangladesh add some unique challenges that are not seen in most parts of the world. In this work, a dataset has been curated to provide a benchmark for Bangladeshi Traffic Sign Detection. The dataset contains 6775 images belonging to 27 different classes of traffic signs containing several challenging samples such as traffic signs of small size, occlusion, illumination variation, blurry condition, etc. reflecting the real-life scenarios. A baseline has been provided by applying the state-of-the-art object detection architectures, where YOLOv5x has been found to be the best performing model with a mAP value of 0.921. A thorough performance analysis has been provided on the curated dataset and further tested on 2000 non traffic sign images to justify the robustness of the model in real-world conditions.

Contents

Acknowledgement

Abstract **i**

1 Introduction **1**

1.1 Overview 1

1.2 Motivation 1

1.3 Research Challenges 2

1.4 Problem Statement 2

1.5 Contribution 3

1.6 Organization of the Thesis 3

2 Literature Review **4**

2.1 Traditional Object Detection Approaches 4

2.2 Deep Learning Based Approaches 5

2.2.1 Two Stage Algorithms 6

2.2.2 One Stage Algorithms 8

2.3 Publicly Available Datasets 11

2.3.1 Belgium Dataset 11

2.3.2 LISA Dataset 12

2.3.3 GTSDB Dataset 13

2.3.4 Tsinghua-Tencent 100K Dataset 14

2.4 Existing Techniques For Traffic Sign Detection 15

3 Dataset Preparation **17**

3.1 BRTA Guidelines 17

3.2 Data Collection 18

3.2.1 Challenging Images 20

3.2.2 Non Traffic Sign Images 21

3.3 Data Statistics 22

3.4 Final Data Preparation 23

3.5 Data Annotation 24

4 Result and Discussion 25

4.1 Experimental Setup 25

4.2 Evaluation Metric 26

4.2.1 Precision 26

4.2.2 AP (Average Precision) 26

4.2.3 Recall 26

4.2.4 mAP 27

4.2.5 F1-Score 27

4.3 Performance of the state of the art models 27

4.4 Error Analysis 29

4.5 Performance of Models on Challenging Images 31

4.6 Performance of Models on Background Images 32

5 Conclusion 35

5.1 Summary 35

5.2 Limitations 35

5.3 Future Work 36

References 37

List of Figures

1 Traditional Object Detection Based Technique, courtesy of [1] . 5
2 General Pipeline of Deep Learning Based Object Detection [2] . 6
3 Architecture of Faster RCNN, courtesy of [3] 8
4 YOLO architecture [4] 9
5 Bounding box creation in YOLO[4] 9
6 Model Scaling of EfficientNet [5] 10
7 Some example images from the Belgium dataset. [6] 11
8 Some example images from the LISA dataset. [7] 12
9 Some example images from the GTSDB dataset. 13
10 Some example images from the Tsinghua-Tencent 100K dataset. 14
11 Example of Bangladeshi traffic sign according to BRTA 18
12 Some example of Bangladeshi traffic sign in different classes of the curated dataset 19
13 Example of blurry traffic sign 20
14 Example of different lighting condition 20
15 Example of small traffic sign 21
16 Example of occluded traffic sign 21
17 Non traffic sign 22
18 Number of images in each class in the Dataset 23
19 Number of images in each class in the final Dataset 24
20 Example of Bangladeshi traffic sign in different classes 24
21 precision and recall curve on yolov5x 28
22 Precision-Recall and F1 curve on yolov5x 28
23 Confusion matrix of yolov5x 29
24 Example of similar types of signs 30
25 Example of similar types of signs 30
26 Comparison of Different Models (Small TS) 31
27 Comparison of Different Models (Night & Blurry TS) 32
28 False detection in non traffic sign dataset 33

List of Tables

1 Scalling Config for EfficientDet d0 to d7, courtesy of [5] 10

2 Model Evaluation 27

3 False detection in non traffic sign dataset 34

Chapter 1

Introduction

1.1 Overview

Object detection and classification is one of the core parts of computer vision [8]. Detecting and classifying facial expressions [9], sign language [10], vehicles and license plates [11], traffic signs [12], road objects for autonomous driving [13], etc, are some of the application areas of this domain. Promising results have been found in these application areas over the last two decades. The introduction of deep learning has boosted the computer vision domain by a great deal [14]. As a result, a significant increment in terms of speed and accuracy is seen in simultaneous object detection and classification from real-time videos. However, the detection and classification of distant objects having complex and noisy backgrounds often make the process harder [15]. Traffic signs aid the drivers by providing them important information to take proper actions while driving. The sign may sometimes be very significant like "School" which should lead to an action of slowing down the speed. So following traffic signs is a must for safe driving. Even laws are adopted to enforce people to follow traffic signs properly. So an automated method of traffic sign detection and recognition can be very crucial for drivers to follow the rules and hence avoid road accidents. But various challenges are encountered while attempting to solve this problem [16].

1.2 Motivation

Road accidents cost a lot of lives every year in Bangladesh and around the world [17, 18]. A report from WHO in 2021 said, 1.3 million people died as a result of road accidents [19]. Failing to follow the different traffic signs on the roads is a major cause behind this catastrophe. Traffic signs generally take a very small porting of the scene. A lot of other similar types of objects in the road like posters, billboards along with complex backgrounds sometimes make it very difficult to identify it. In spite of the high significance of robust detection and recognition system for Bangladeshi traffic signs, not too many works are found in the existing literature. Besides there is no publicly available benchmark dataset containing Bangladeshi traffic sign images reflecting different real-world scenarios.

1.3 Research Challenges

One of the challenges of traffic signs is that they are typically contained within only 0.2% of the entire image. Earlier research focused on creating a well organised and balanced traffic sign detection and recognition benchmark dataset and implementing different models to find the best result. German Traffic Sign Detection Benchmark (GTSDB) [20], German Traffic Sign Recognition Benchmark (GTSRB) [21], Belgium Dataset[6], Tsinghua-Tencent 100k dataset[22], Laboratory for Intelligent and Safe Automobiles (LISA) [7] are the widely used datasets in this domain. Various detection and classification methods are used on those datasets and some of them achieved significantly good results. But when real world challenges are portrayed, the results can vary a lot. Creation of such a dataset with wide variety of conditions and achieving high performance is a challenging task.

In our research we found out that no existing public dataset covers all the most common traffic signs of Bangladesh. There are few works with on Bangladeshi traffic sign detection with self-collected private datasets and do not contain significant amount of data reflecting real-world challenges. Also the publicly available benchmark datasets did not address all the challenging conditions like occlusion, night images, small traffic signs etc.

1.4 Problem Statement

This research addresses the necessity of a publicly available traffic sign dataset that can be used in the making of a robust traffic sign detection and recognition system. Besides, the research also focuses on finding the best suited model for the curated dataset that can handle the challenging conditions most effectively.

1.5 Contribution

The contribution of our research works are:

1. We have curated a benchmark dataset of Bangladeshi traffic signs covering different weather & lightning conditions, occlusion, blurriness, small traffic signs, night images, etc.

2. We also presented a comparative analysis with the state-of-the-art object detection models and mentioned the best suited one for the dataset.

1.6 Organization of the Thesis

The remainder of this article is organized as follows:

Chapter 2 gives a Literature review discussing the different approaches and techniques used in the literature of object detection over the years. This section also describes some of the publicly available datasets. Chapter 3 provides detailed process of our data collection phase and current state of our dataset. Chapter 4 presents the experiments we conducted and result analysis. Chapter 5 gives an overall conclusion of our thesis and discusses our limitations and future plan of work.

Chapter 2

Literature Review

Object detection is a process of computer vision that localizes objects within an image. It mainly consists of 2 stages. At first, bounding boxes are created around the objects within an image, then the objects are labelled. There are two types of object detection techniques, traditional and deep learning based approaches.

2.1 Traditional Object Detection Approaches

The traditional object detection based approaches mostly processes various features of an object, for example, color histogram, pixels, edges, etc [12, 23]. At first the background is distinguished from the foreground of the object in the image. Then these features of the foreground objects are manually extracted and fed into a classification model to localise the objects in the image.

The traditional Machine learning based object detectors mostly focused on the color based features of traffic signs [12]. As traffic signs generally have red or blue borders, the edges of objects are detected utilizing these specific colors. For this purpose Histogram of Oriented Gradients (HOG) transformation was widely used in many literature. [24] converted the original image into HSV image and then applied threshold filters and contour to detect the object. Then it validated through inverse threshold filter. After that it detected the object using Convolutional Neural Network (CNN). [25] used HSV to extract blue and red pixels and used generalized HOG transform to detect the object. After that it used CNN classifier to classify that object.

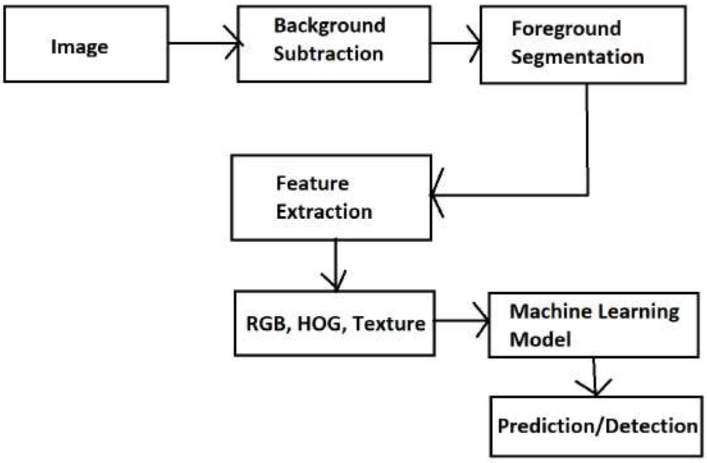


Figure 1: Traditional Object Detection Based Technique, courtesy of [1]

2.2 Deep Learning Based Approaches

The deep learning based object detectors use convolutional neural network to perform an unsupervised and end to end object detection. Multiple complex features are extracted automatically through convolutional neural network. CNN consists of convolutional layers, activation function, pooling layer, fully connected layer and finally a classifier. The convolutional layer learns the specific filters which create a pattern and eventually an object in the image. The pooling layers are in between two convolutional layers. A pooling layer reduces the parameters of the output of a convolutional layer. Thus it decreases the computational complexity. The activation function does linear or non linear operations on the input data. It determines which neuron will be triggered. In the fully connected layer, all the neurons are connected to each other. This layer reduces the input data based on the classes that are being trained. The classification layer contains a softmax function. The softmax function converts the output of the fully connected layer into probability scores for different classes.

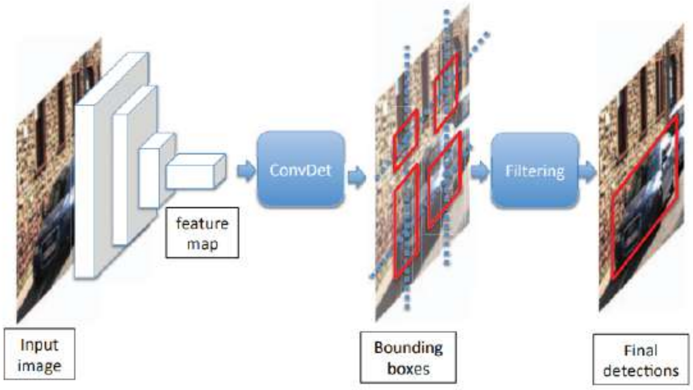


Figure 2: General Pipeline of Deep Learning Based Object Detection [2]

2.2.1 Two Stage Algorithms

In two stage models, the first stage is used for extracting the regions of objects and the second stage is used to classify the object. R-CNN(Region Proposal Network), Fast R-CNN [26], Faster R-CNN [27], Mask R-CNN [28], FPN(Feature Pyramid Network) [29], are some state of the art object detection algorithms. They generally have high localization and detection accuracy. [30, 27] used Faster R-CNN for detection purpose and for classification they [30] used VGG-16, VGG_CNN_M_1024 with the detection model.

Faster R-CNN ResNet: Faster R-CNN ResNet [27] detects objects in two steps: region proposal and classification. At first, with the help of Region Proposal Network (RPN), the model creates around two thousands of regions. R-CNN and Fast R-CNN used selective search algorithm for this purpose. But the algorithm was very slow. Faster R-CNN solved the problem with RPN. The RPN creates 9 anchors by default in the image. It filters out anchors with no object based on their objectness score. In the second stage, Thus it generates a probability that the anchor is an object. The probability score is one of the outputs of the RPN. The second output of RPN is the bounding box regression. It is done to better fit the object's prediction. Creating region proposals, the ROI-pooling layer creates feature vectors from each of the regions proposals with shared computation power. SVM (Support Vector Machine) classifier takes the feature vector as input and determines if the region has an object or not. If the anchor has an object, then it is the foreground. Otherwise, the anchor is background. The overall loss of a RPN is the combination of the classification and regression loss.

The Residual Neural Network (ResNet) is used for classification of the object. ResNet-101 has 101 layers while ResNet-152 has 152 layers. ResNet-152 performs better than ResNet-101 as the former one has more layers than the later. During the classification process, the anchors labelled as background are not processed. In the regression process, the location of the final bounding box is the output. During training, all the anchors are categorized into foreground and background. The anchors that overlap the ground truth object with IoU (Intersection Over Union) value greater than 0.5 are considered as foreground while the rest are considered as background. The RPN uses Binary Cross Entropy to calculate the classification loss. To calculate classification loss, all the anchors are selected. To calculate regression loss, the foreground anchors are selected. Also the anchors that are close to the ground truth object are processed to calculate the difference value needed to become a foreground object. As a lot of anchors can be overlapped, there will be a lot of processing if all the anchors are selected. To eradicate the problem, an algorithm called Non Maximum Suppression is used. The NMS algorithm takes all the anchors as inputs and discards the anchors which are less than the threshold value of IoU. Thus the number of anchors are decreased. After applying NMS algorithm, top N number of anchors are selected from the sorted list.

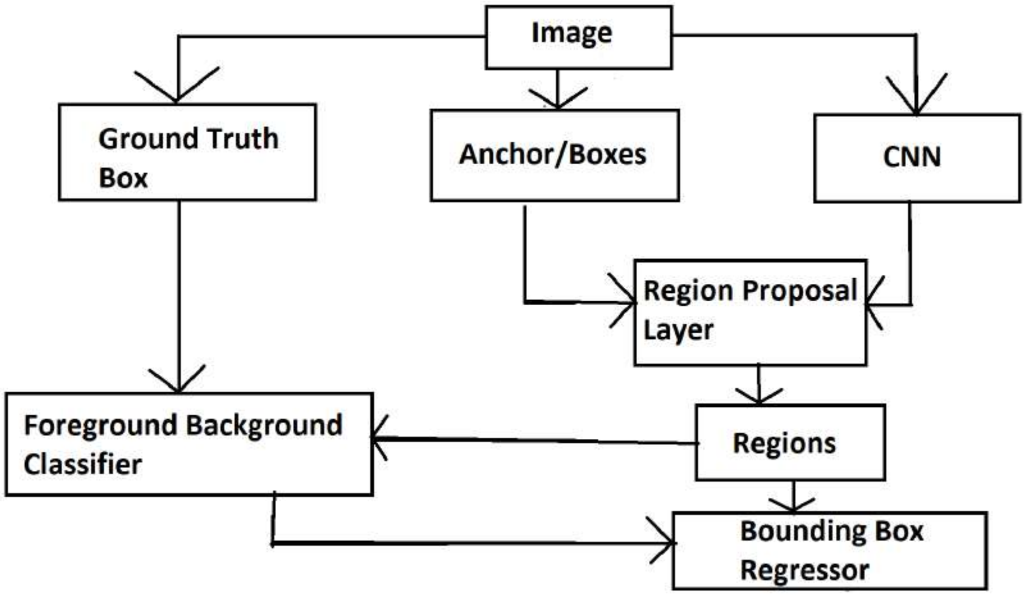


Figure 3: Architecture of Faster RCNN, courtesy of [3]

2.2.2 One Stage Algorithms

One stage detection algorithms detect and classify objects in one pass of the image. Family of YOLO(You Only Look Once), SSD(Single Shot Detector), EfficientDet [31] are the one stage algorithms. They are faster in speed with high accuracy. [32] Used YOLOv3 with Darknet-53 in its backbone for both detection and classification.

YOLO: YOLO is a one-stage object detection model. It does detection and classification at the same time. This algorithm divides each image into N numbers of grid cells and each grid cell has equal size. The model checks each of the grid cells and finds out the targeting object to create anchor boxes. It creates a lot of duplicate bounding boxes because the same object is detected on multiple grids. To solve this problem, YOLO uses Non Max Suppression. The NMS algorithm removes the bounding box with low probability.

YOLO is faster than other algorithms but it struggles to detect small objects in the image. For Small objects in images, slower deep learning models like Fast RCNN or Faster RCNN work better than YOLO.

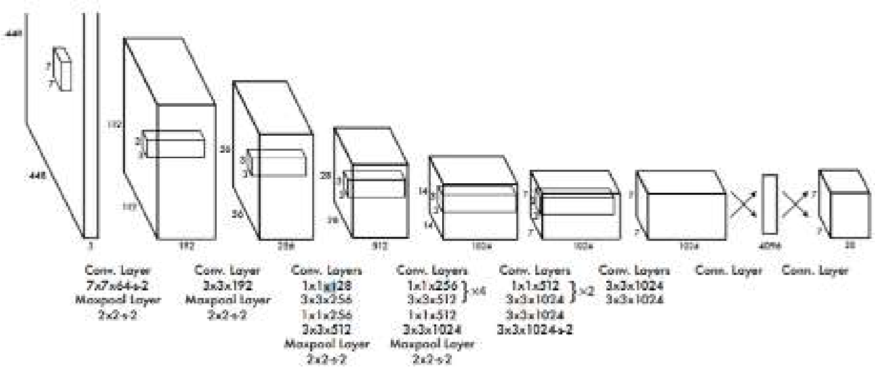


Figure 4: YOLO architecture [4]

YOLO has 24 convolution layers and 2 fully connected layers. YOLO has released several versions such as YOLO, YOLOv2,YOLOv3, YOLOv4 and YOLOv5. For experiments on our dataset we use YOLOv5s and YOLOv5x.

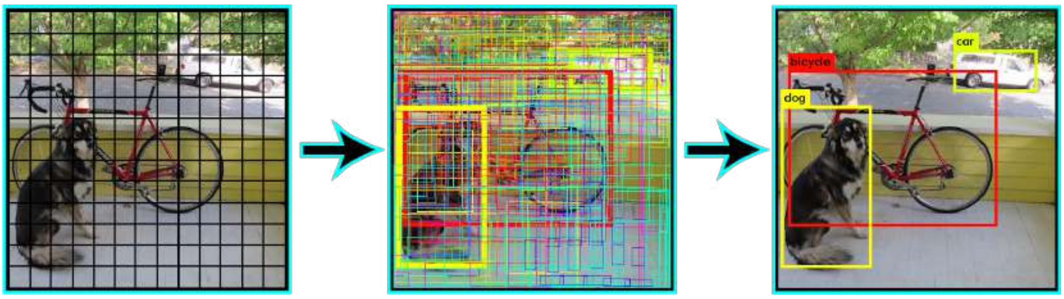


Figure 5: Bounding box creation in YOLO[4]

EfficientDet : EfficientDet [31] was proposed in 2019, by a team of Google Research. Typically the scaling of a neural network is done on width, height or resolution of the baseline network. But using all of them to scale a network, it gives the optimum performance. This scaling process is called compound scaling. The baseline network of the EfficientDet is called Mobile Inverted Bottleneck Convolution. EfficientDet follows the 1 stage detection paradigm. It detects and classifies input images in a single stage. It is comprised of EfficientNet and BiFPN layers. EfficientNet is the backbone of the EfficientDet architecture. EfficientNet is pretrained on ImageNet dataset. Using the compound scaling, upto EfficientNet B7 has been proposed with greater performance.

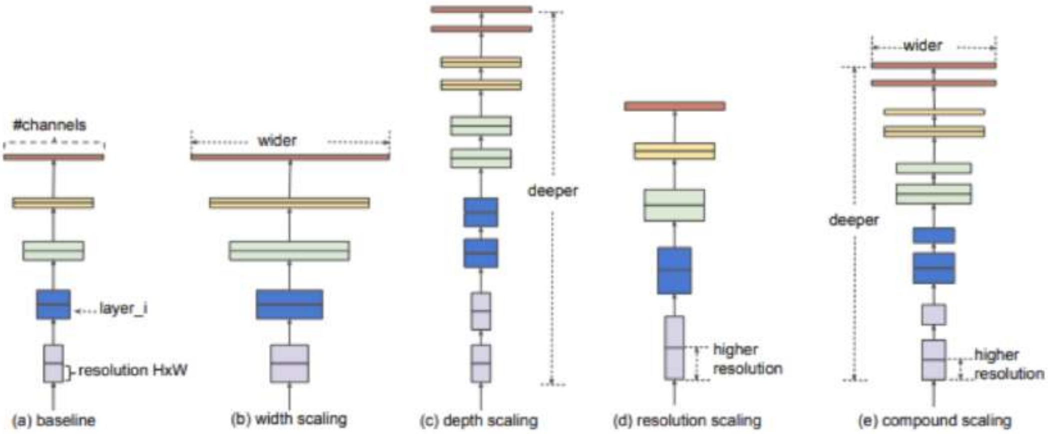


Figure 6: Model Scaling of EfficientNet [5]

Keeping the target memory and target flops intact, EfficientNet scales the depth, width and resolution of the input to optimize the performance than the previous ConvNets. BiFPN layer is used for feature fusion. This layer takes features as inputs from the backbone layer, EfficientNet. Then the features fused together in a top down and bottom up bidirectional feature fusion approach. Then the fused features are sent to a predictor class and box network to produce the class name and bounding box respectively.

Table 1: Scalling Config for EfficientDet d0 to d7, courtesy of [5]

	Input size	**Backbone Network**	**BiFPN**		**Box/class**
			#channels	**#layers**	**#layers**
D0	512	B0	64	3	3
D1	640	B1	88	4	3
D2	768	B2	112	5	3
D3	896	B3	160	6	4
D4	1024	B4	224	7	4
D5	1280	B5	288	7	4
D6	1280	B6	384	8	5
D7	1536	B6	384	8	5
D7x	1536	B6	384	8	5

2.3 Publicly Available Datasets

2.3.1 Belgium Dataset

Belgium Dataset [6] was published in 2011. There are a total number of 1,45,000 images with 62 different types of traffic signs. 13,000 of the images are annotated and their size ranges from 100×100 to 1628×1236 pixel. Their contemporary public traffic sign datasets had less sign types than the Belgium dataset. Moreover, other datasets focused on mainly highway traffic signs while Belgium dataset collected images from smaller roads. Thus the images contain smaller and challenging traffic signs.

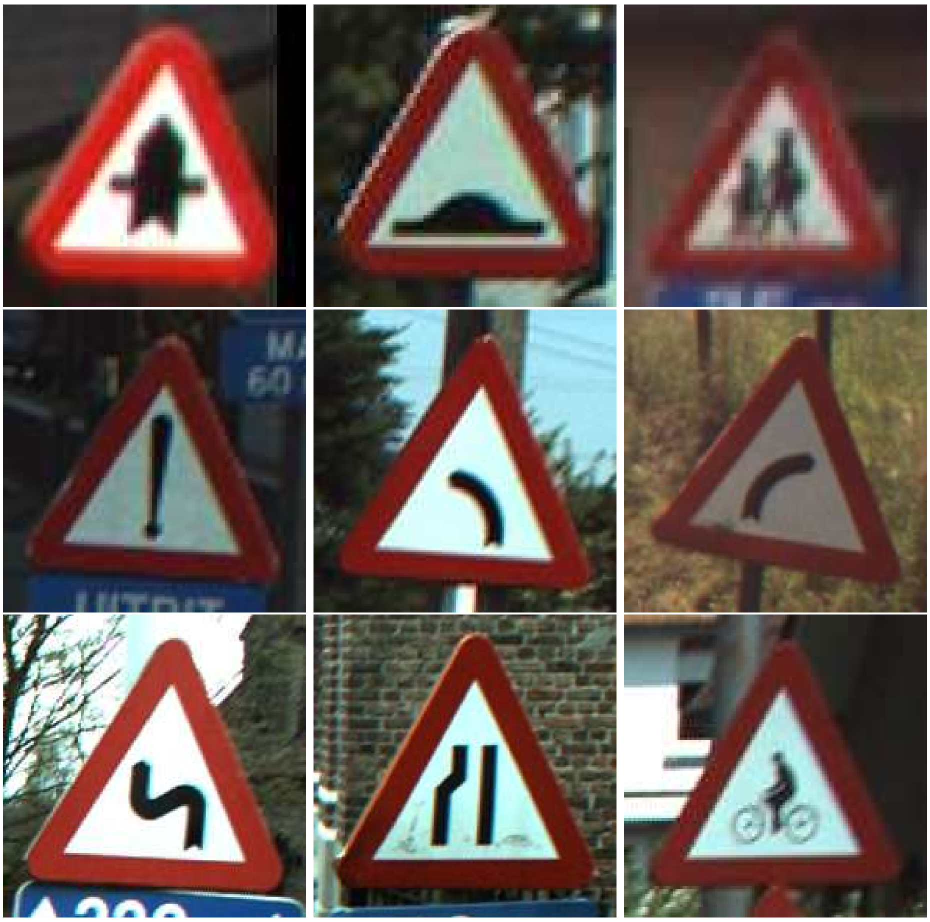


Figure 7: Some example images from the Belgium dataset. [6]

The authors mentioned 2 ways of traffic sign detection, One is selective extraction of windows of interest followed by their classification and the other one is exhaustive sliding window based classification. The research also suggests that combining them both is a good idea to achieve higher performance.

For selective extraction, the paper proposes an off-line learning approach to select features and corresponding thresholds automatically instead of choosing them manually. The proposed multi-view 3D localisation ensured better performance than single view detection method. Besides that, an efficient evaluation for linear discrete Ada-Boost like classifiers is proposed without trading off the performance.

The multiview 3D localisation model had 95.3% detection accuracy and 97.0% recognition accuracy with the Belgium dataset.

2.3.2 LISA Dataset

The LISA (Laboratory for Intelligent and Safe Automobiles) dataset [7] published in the year of 2012 proposed a benchmark dataset for traffic sign detection. It contains 6610 American traffic signs which are distributed in 49 classes and the images range from 640×480 to 1024×522 pixel in size. It includes videos of all the annotated images and all images are annotated.

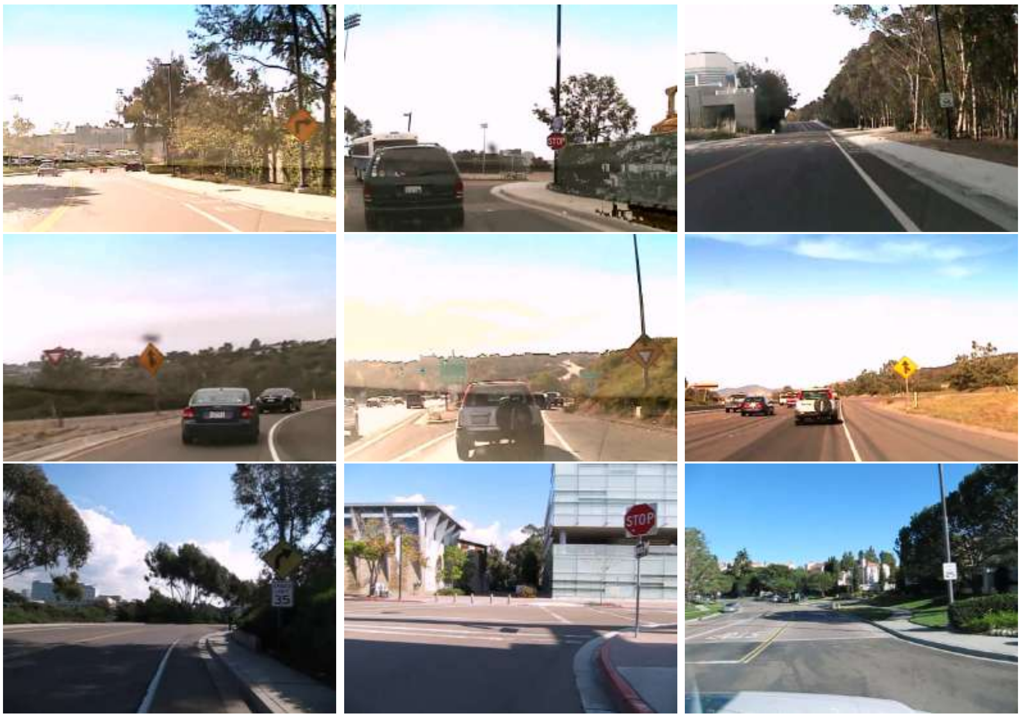


Figure 8: Some example images from the LISA dataset. [7]

The contemporary public American traffic sign datasets were old and in those datasets, traffic sign types varied a lot. Many of them were faded, unclear, and had low contrast. The contemporary benchmark datasets were GT-

SRB, KUL (Belgium Dataset), STS (Swedish Dataset) [33], RUG (Netherlands Dataset) [34] and Stereopolis (French Dataset)[34]. As none of them could provide a benchmark for American traffic sign detection, the LISA dataset came up with a solution.

2.3.3 GTSDB Dataset

German Traffic Sign Detection Benchmark [20] contains a total 900 traffic signs, collected from several tours. They capture videos of real time scenarios like urban, rural, highway and in several weather conditions. The final dataset is split into a training set containing 600 images and a testing set containing 300 images. Then all the images are annotated with rectangular regions of interest (ROI).

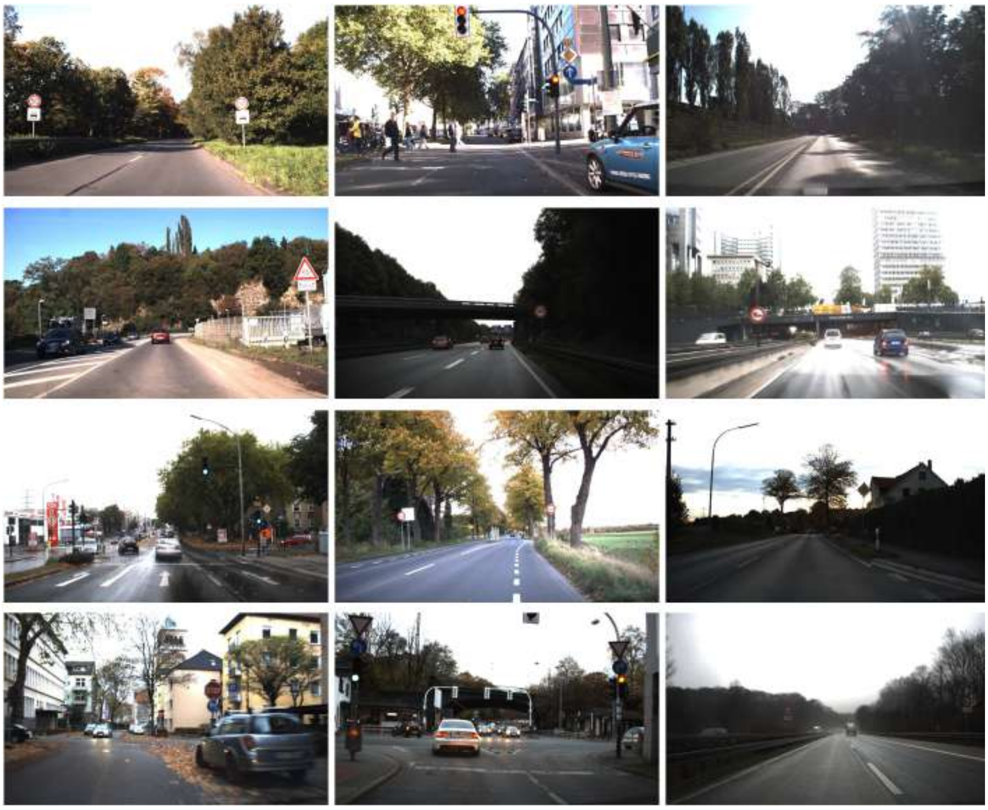


Figure 9: Some example images from the GTSDB dataset.

Current state of the art methods achieve above 90% precision and recall in the dataset [20]. But it only detects three major categories of traffic signs: prohibitive, mandatory and danger signs. In GTSRB most of the region of the image contains the traffic sign and there is no negative sample. So it is easy to classify all the images.

2.3.4 Tsinghua-Tencent 100K Dataset

[22] proposed Tsinghua-Tencent 100K (TT100k) benchmark dataset to simultaneous detection and recognition. It contains 100k images of them only 10000 images contains traffic sign with 30000 traffic sign instances in 100000 classes and others 90000 images does not have any traffic sign instances. It provides a large number of negative samples to make the detection process difficult. All the images are extracted from Tencent Street Views.

In the training process, they ignored classes which contains less than 100 instances. To make the dataset balance, they use augmentation and make all the 45 classes into 1000 instances.

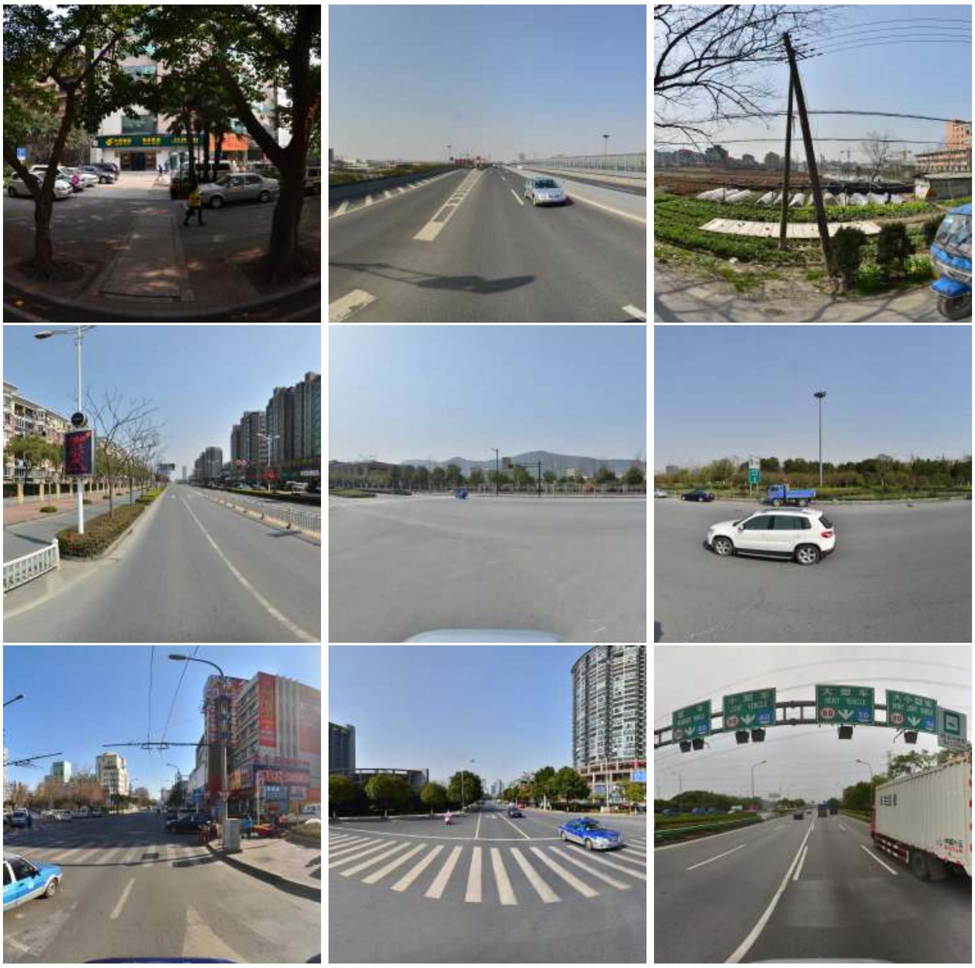


Figure 10: Some example images from the Tsinghua-Tencent 100K dataset.

A multi class network was proposed for traffic sign detection and simultaneous detection and recognition. It achieved 84% accuracy and 94% recall for detection and recall 0.91% and accuracy 0.88% for simultaneous detection and recognition. But there is no night images in their dataset. And night images are hard to detect in any model. In real world scenarios distant images are more difficult to detect. So we create a Benchmark dataset to simultaneously detection and recognition of traffic signs.

2.4 Existing Techniques For Traffic Sign Detection

Over the last few years, several techniques are seen to be applied in this domain . [35] proposed two new networks for classification and detection and they are ENet and EmdNet respectively. The classification network is similar to the LeNet [36] network architecture and they presented the best combination of hyperparameters after conducting numerous experiments. In the detection part they combined depthwise separable convolutions and multi-scale operations. VGG-16, VGG_CNN_M_1024 and ZF were trained using a self collected Chinese dataset by [30] and they found that ZF model got the highest detection accuracy.

[37] used Viola-Jones framework along with hybrid pipeline combining Machine Learning and Deep Learning classifiers for traffic sign recognition. A Region Proposal Network based on YOLOv3 architecture is used by [38] which proved to locate small traffic signs better than the standalone YOLOv3 network. They achieved that by adding one extra layer to the decoder network than YOLOv3. Besides that based on the logo of traffic signs, they applied data augmentation.

One of the major challenges is classifying similar traffic signs efficiently. For doing so [39] used multi-scale attention method. A multi-scale cascaded R-CNN was proposed for dealing with small sized traffic signs. Also hard negative samples were mined and distributed equally to each classes. For detection purpose, [40] used HSV color space and for the classification part, an improved LeNet-5 model architecture was proposed where Gabor Kernel is initial kernel of the network. [41] used the Single Shot Detector algorithm by adding multi-feature fusing in it which enhanced the detection capability of small traffic signs. Color based feature extractors are used at first by [42].

Then they used Bilateral Chinese Transform for detecting circular shaped traffic signs and Vertex and Bisector Transform for detecting rectangular traffic signs. [43] calculated the probable position where a traffic sign can be found and applied YOLO model with a region of interest based approach. With this they achieved a fast processing time with good accuracy. An independent attention detection model with multi-scale detection algorithm is provided in [44] to achieve better performance than other approaches in TT100k dataset.

Although there has been a lot of works in traffic sign detection in recent years, the field of Bangladeshi traffic sign detection is relatively unexplored. In [45] they collected 759 daylight images from different districts in Bangladesh and incorporated 16 road sign classes. Then they used Single Shot Multibox Detector (SSD) for detection and Recognition at the same time. They also used Convolutional Neural Network for recognition purpose. They achieved 76.52% detection and 86.23% recognition accuracy using SSD and 80.26% accuracy using CNN model. In 'Narrow Bridge' road signs both models achieved 0 precision and 0 recall. On the other hand, SSD showed 100 precision and 50 recall score for 'School Ahead' class but the CNN model produced 0 precision and 0 recall. They showed that in their dataset SSD worked better than CNN model.

Distance to Borders (DtBs) vector and Artificial Neural Network (ANN) was used on a self-collected dataset consiting of 110 images in [46]. DtBs is applied to detect the multi-colors and multi-shapes road sign and to recognize the classes of road signs, ANN is applied. The authors achieved 94.87% detection accuracy and 92.79% recognition accuracy which is better than [47] and [48].

[45, 46, 49, 50, 51] worked on Bangladeshi Road sign dataset, but all of them used a small amount of images in their dataset belonging to only a few classes. None of those datasets are publicly available. The works mostly were based on color based and edge based machine learning algorithms which are most likely fail to perform well in challenging conditions. Therefor, a publicly available benchmark dataset reflecting real-life challenging scenarios can go a long way to build robust traffic sign detection systems.

Chapter 3

Dataset Preparation

This chapter explains about the process of curating our dataset named 'Bangladeshi Traffic Sign Detection Benchmark (BdTSDB)'. The data collection process, inclusion exclusion criteria, annotation process, etc have been discussed in detail. The class distribution of the proposed dataset has also been discussed.

3.1 BRTA Guidelines

Bangladesh road transport authority under the ministry if communication has provided a traffic sing manual[52] which imposes a technical guideline of every aspects of traffic signs. They have categorized all the traffic signs in 3 groups, Regulatory Signs, Warning Signs, Information Signs. The regulatory signs tell the drives what actions they must do and what they must not do. They are generally circular in shape. 'No Use of Horn', 'Keep Left', 'One Way Traffic' are some of the examples of regulatory signs. The warning signs are there to warn the drivers about upcoming difficulties of the road and they are generally triangular in shape. Some of the mandatory signs are 'Traffic Merges From Left', 'Sharp Bend to the Right', 'Location of Railway Crossing'. Information signs provide useful direction to the drivers that can lead them to their destination and some of them are 'Pedestrian Crossing', 'Lane Ahead for (cycles and rickshaws), 'Toll Road or Bridge'. There are 40 regulatory signs, 57 warning sings and 35 information signs. They have provided detailed description of each and every sign. The descriptions include dimension, color, description, application, location and variation of that sign. In our dataset we only included the traffic signs that are present in the BRTA's guideline.

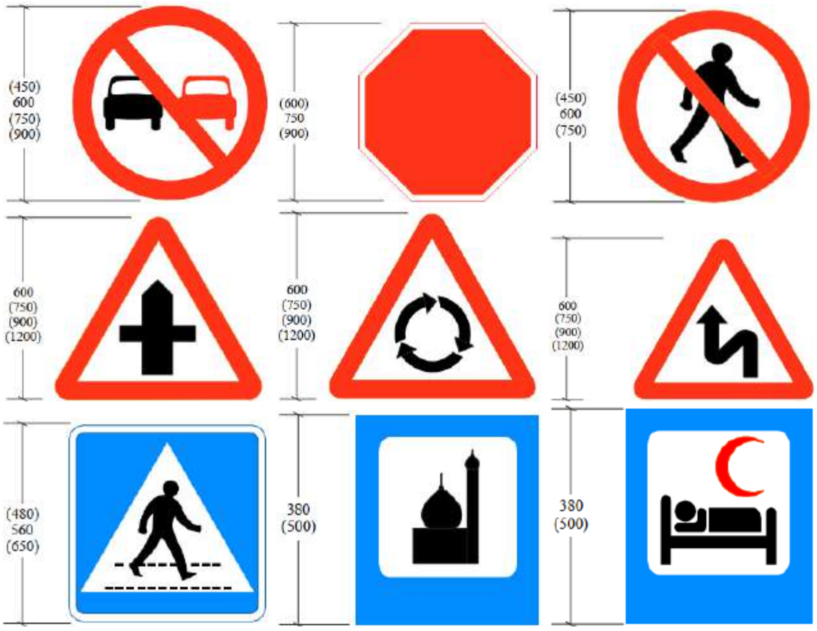
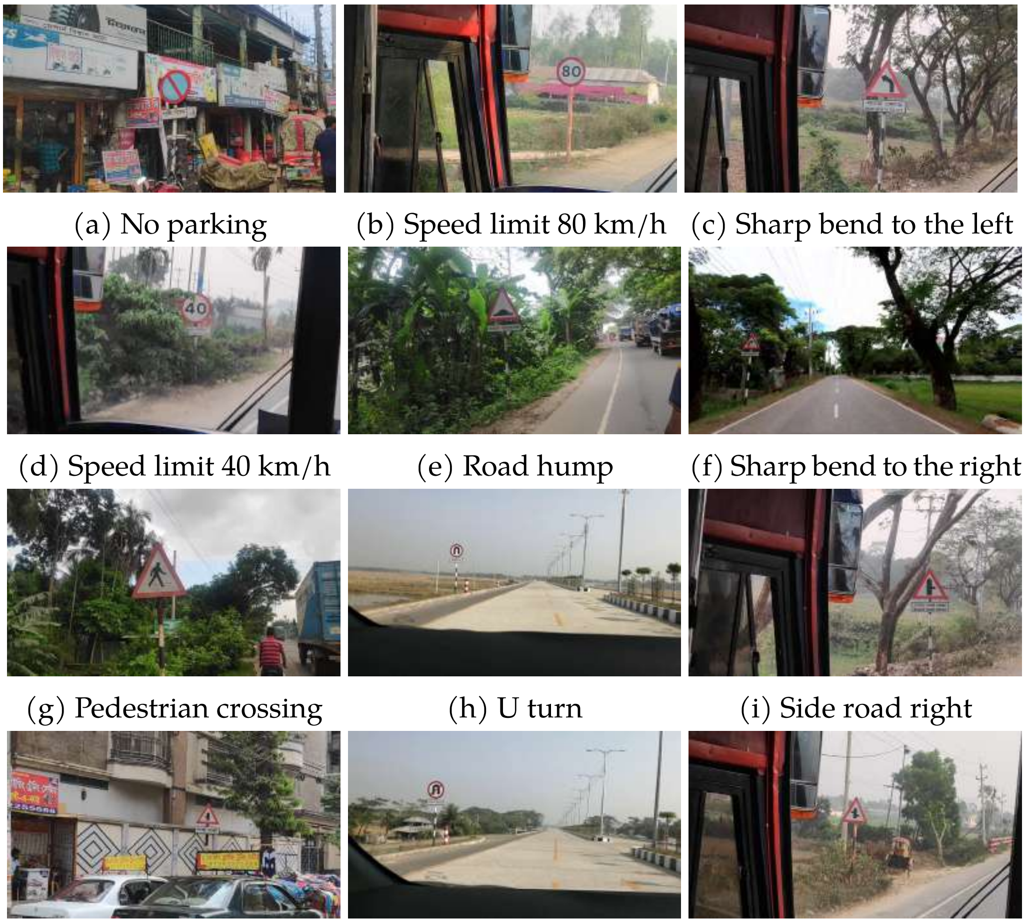


Figure 11: Example of Bangladeshi traffic sign according to BRTA

3.2 Data Collection

The process of collecting thousands of images having traffic signs is an exhaustive one. One of the ways of doing it is to collect street videos and extracting those frames which have traffic sign in them. We applied this process and for that we gathered videos from 2 sources. One is going to different parts of Bangladesh and capturing videos from the street by ourselves and the other one is searching and collecting videos from youtube channels that can serve our purpose. We have collected a total of 197 videos. Among them 163 videos were collected by ourselves and the rest 34 videos were collected from youtube. The videos that were collected by our selves are from different roads of Dhaka city, Sylhet, Dhaka Chittagong highway and a few other places. The videos collected from youtube are mostly from highways and rural areas. Thus we ensured that we have videos from different geographical locations. Different types of vehicles were used to collect those videos.

We used 4 different types of mobile devices to record videos from different roads in Dhaka and outside Dhaka. The fps value of the videos were 30 and the resolution of the videos ranged from 1280x720 to 1920x1080. Then the challenge was to extract frames from those videos having traffic sign in it. As some of those videos were long and most of the frames did not contain any traffic sign, it was a challenging task to make that process efficient. For that at first we we took the Faster R-CNN Inception ResNet v2 model, pre-trained on the GTSDB dataset and passed some of those videos in it to find the frames with traffic signs and saved those frames. But we found that the model missed a lot of traffic signs and also had quite a few false positives. Then after trying some other approaches, we finally got stuck to an approach in which we went thorough a video and noted the timestamps when we found traffic sing. Then using a script we extracted the frame of that timestamp and 5 frames before and after that timestamp with an interval of 10 frames. Then we went through all the frames and looked for any frames that have no traffic sing in it and discarded them.



(a) No parking (b) Speed limit 80 km/h (c) Sharp bend to the left

(d) Speed limit 40 km/h (e) Road hump (f) Sharp bend to the right

(g) Pedestrian crossing (h) U turn (i) Side road right

(j) Side road left (k) No u turn (l) Staggered junction

Figure 12: Some example of Bangladeshi traffic sign in different classes of the curated dataset

The vehicles that were used to record road videos were of different speed ranges. We extracted frames with a frame gap of 10 for high speed vehicles and with a frame gap of 5 for low speed vehicles. After that we put them into their predefined respective classes. Finally we got 9689 images in total that were distributed in 53 classes.

3.2.1 Challenging Images

To portray the real world scenario our dataset contains images with different weather conditions such as rainy days, different lighting conditions, blur, occlusion, day-light, night and different orientations.

Blurriness is generally occurred due to the speed of the vehicle. Also inappropriate shutter speed of camera may make the photo blur. We have been able to include a good number of blurry images in our dataset. After inspection of the dataset we have identified 347 blurry images. Figure 13 shows some of the examples of blurry traffic signs from the prepared dataset.

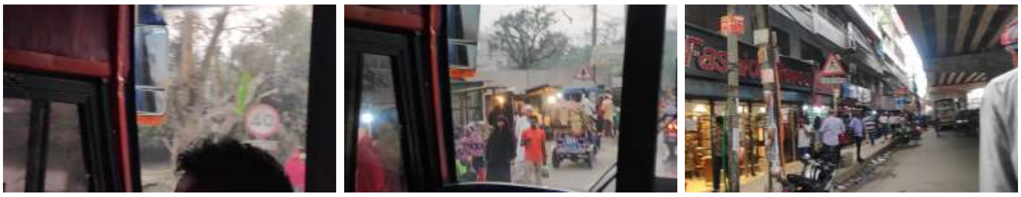


Figure 13: Example of blurry traffic sign

Different lighting conditions can have an impact on the visibility of an object. For example, the flash light of the vehicle, different phases of a day (photopic or daylight, mesopic or twilight, scotopic or night), background luminance creates impact on the visibility of the traffic signs. We have found 374 images with lighting condition in our dataset which also included night images. In figure 14 some of the instances of lighting conditions are presented.

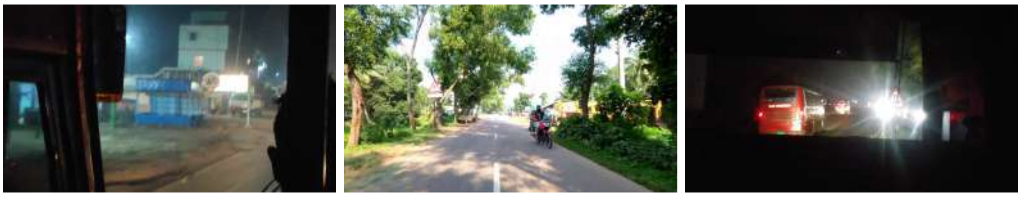


Figure 14: Example of different lighting condition

One of the challenges of object detection is to efficiently detect small objects.

Many research have been done considering the detection of small objects. If the distance of the vehicle is far from the traffic sign then the traffic sing appears as a small object. Even in the ideal situation, traffic signs contain a small portion of the entire frame. So when the sign gets even smaller, it become very challenging to correctly detect and classify that sign. We can see some examples of small traffic signs in figure 15 from the 697 images obtained from the prepared dataset. We have found 697 images containing small traffic signs in it.

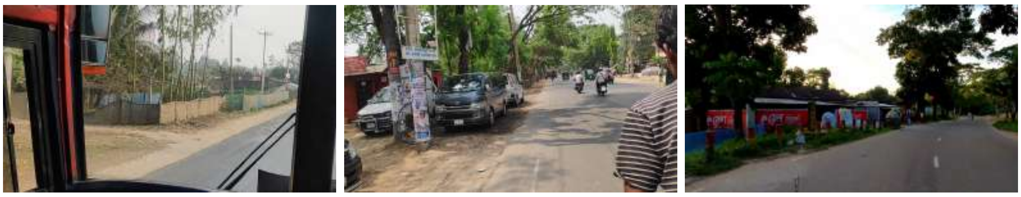


Figure 15: Example of small traffic sign

Occlusion is one of the nuisance for traffic sign detection that is often seen in real world scenarios. We have found images where the traffic sign is occluded by other vehicles, by leaves and branches of trees. We have found a total number of 178 of those images. Some of the occluded samples of the dataset is shown in figure 16.

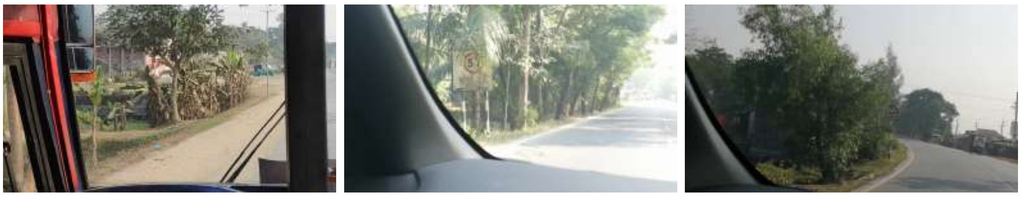


Figure 16: Example of occluded traffic sign

We have also included images from rainy condition which made our dataset even more diversified.

3.2.2 Non Traffic Sign Images

In real world scenes we encounter a lot of traffic sign like objects which are not traffic sign. These signs typically are the backward faced traffic signs, banners, posters, billboard and different shape of objects that can often be matched with a particular traffic sign. Distinguishing them from the actual traffic signs is a huge challenge. [22] addressed this problem by adding a huge amount of non traffic sign images in their evaluation set. They tested those images using their proposed model. For that we collected 2000 non traffic sign images from youtube and our collected recorded videos. The frames of those videos are extracted at random while ensuring that each frame contains a different scene. In figure 17 a few of those samples can be seen. These non traffic sign images include objects which may look like traffic signs but they are not. These images are also tested with the models that we trained with our dataset.

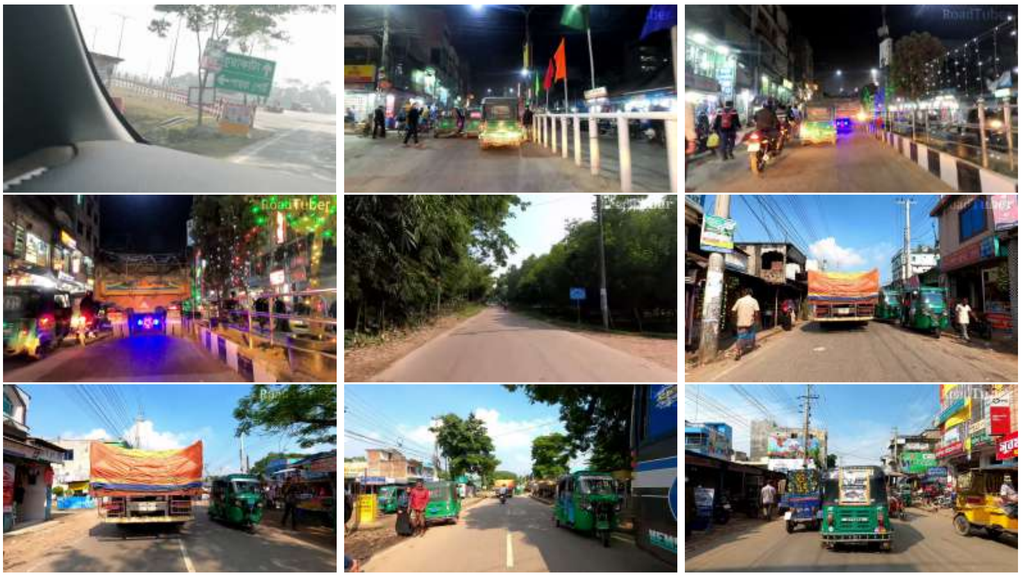


Figure 17: Non traffic sign

3.3 Data Statistics

After extracting frames from 197 videos, we found 9689 images and distributed them in 53 classes. The most number of images were in 'Pedestrian Crossing' class with 1464 images in it as it is the most common traffic sign seen in the road. After that among the most common traffic signs we have 'No Overtaking' having 957 images, 'School' having 888 images, 'speed limit 40 km/h' having 649 images in it. Some of the rare traffic signs seen in Bangladeshi roads are 'Keep Right', 'Hairpin Bend To Right', 'Major Road Ahead (Crossroads)', 'No Vehicles Over Maximum Gross Weight Shown'.

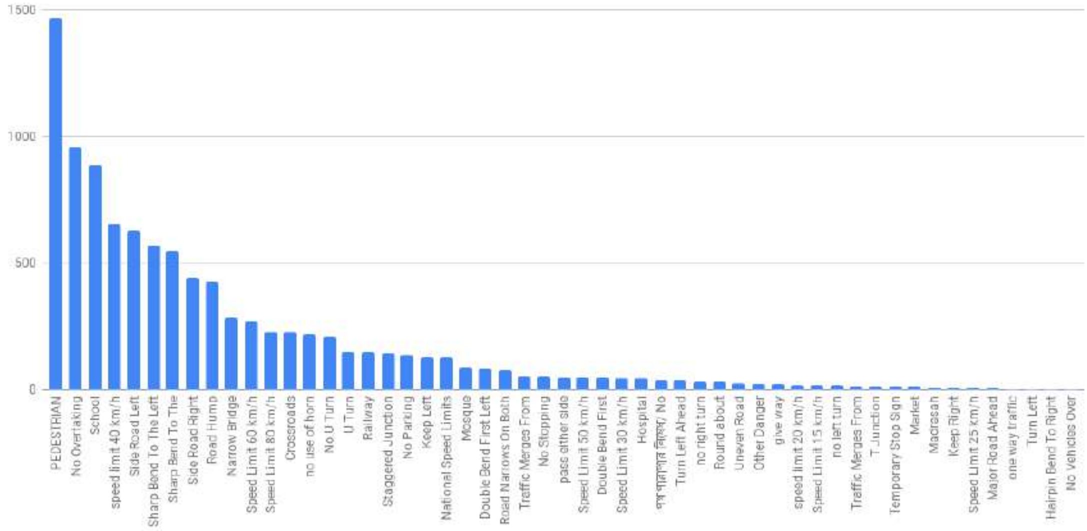


Figure 18: Number of images in each class in the Dataset

3.4 Final Data Preparation

After the initial dataset collection, we went through each and every images to filter consecutive frames and discarding images where the traffic sign is too hard to be identified. After the filtering process we got 7204 images. This dataset has some imbalance due to having less number of images in some of the classes. Then for experimental purpose we imposed a constrained and took only those classes that have at least 50 images in them. So finally we got 6775 images for our experimentation that were distributed in 27 classes. Here also 'Pedestrian Crossing' contains the most number of images with 975 images. For splitting the dataset into training and testing set we randomly shuffled each of those 27 classes so that we can tackle the case of consecutive frames going to just one of those sets. Finally we splitted the whole dataset into training and testing set using 80/20 policy. Our training set contains 5260 images and evaluation set contains 1350 images.

While going through the whole dataset we also identified the different challenging images and gathered them in their respective classes. It helped us in our model evaluation process in understanding which model works well on which type of challenge.

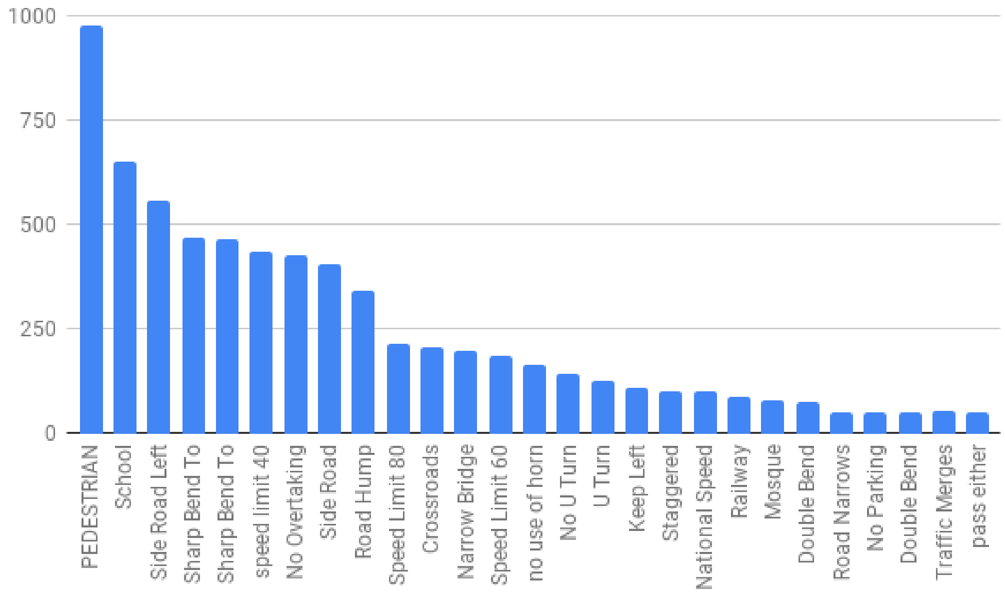


Figure 19: Number of images in each class in the final Dataset

3.5 Data Annotation

After the final dataset preparation, we carefully annotated all the images. We used a rectangular region of interest with their label using the 'LabelImg' annotation tool. We saved the annotated images in Pascal VOC format. This format can easily be converted into other annotaion formats like COCO JSON, Tensorflow TFRecord. The LabelImg tool saves the annotations as .xml files.

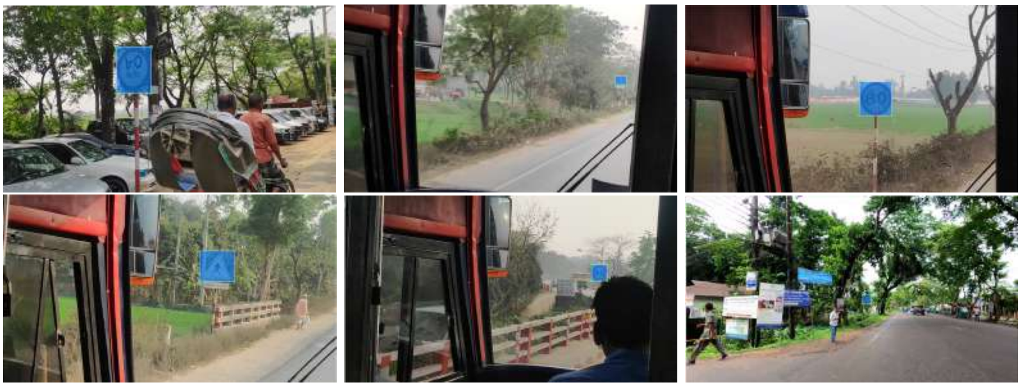


Figure 20: Example of Bangladeshi traffic sign in different classes

Chapter 4

Result and Discussion

4.1 Experimental Setup

After preparing the dataset, we trained and evaluated it with six state-of-the-art object detection models. For experimentation we used the Tensorflow Object Detection API [53] which is a open source framework built on top of Tensorflow. It has support for both Tensorflow 2.x and 1.x. The API provides a model zoo which has a large collection of pretrained models on the COCO 2017 dataset. They have provided the speed, mAP and output of each of those models after training with COCO 2017 dataset. We picked 4 of those models and they are EfficientDet D0, Faster R-CNN ResNet 101, Faster R-CNN ResNet 152, Faster R-CNN Inception ResNet v2. We also worked with YOLOv5s and YOLOv5x which are not included in the API.

We have used Google Colab Pro to conduct the experiments. There we worked with Tesla P100 GPU and 16280MB RAM. Google Colab has both the versions of Tensorflow preinstalled in it. We also did not need to install Tensorflow Object Detection API.

For the pipeline configuration, at first we downloaded the Tensorflow Model Garden from their git repository. Then we downloaded the protobuf libraries and compiled them. We created a label map file in which for each label there is an integer id. Tensorflow uses the label map file for both training and evaluation. As mentioned earlier, we generated *.xml files while annotating with labelImg tool. In the next step we took all those *.xml files and converted them into *.record file. As a result we got two files, train.record and test.record. The TFRecord file format stores data as a sequence of binary records.

For the configuration of the training job, we downloaded the pre-trained models from Tensorflow 2 Detection Model Zoo. We needed to change the pipeline configuration file to make it prepare for our job. We trained those 4 models using different number of steps and each of them had varied convergence rate. We applied cosine decay learning rate, RELU activation function. The IoU threshold used was 0.6 and sigmoid as score converter. For the evaluation metric we used COCO detection metric. We used TensorBoard to monitor our training job progress. For the YOLOv5s and YOLOv5x, we used PyTorch framework.

4.2 Evaluation Metric

4.2.1 Precision

The number of positive classes correctly classified by the model from the total number of positive predicted classes. It measures how accurate the percentage of the model is. It gives a value of 0 to 1 which denotes the percentage of the correct predictions. For example if the model predicts 5 of the samples positive but there were actually 3 positive samples then the precision value will be 3/5.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{1}$$

4.2.2 AP (Average Precision)

Average Precision denotes the area under the precision recall curve. As both the precision and recall value is between 0 and 1, average precision value also resides between 0 and 1.

4.2.3 Recall

The number of positive samples correctly classified by the model from the total number of positive samples. It measures how good the model is in finding all the positive classes. For example if there are a total of 5 positive samples in the dataset and the model could predict 2 of then then the recall will be 2/5. As a result, as the model predicts more samples, the recall value of the model increases.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

4.2.4 mAP

Mean Average Precision calculates the comparison of the ground truth bounding box with the detected bounding box. When we take the average of AP among all the classes, we get the mean average precision. Generally AP and mAP are mentioned interchangeably. Mostly COCO mAP is used in evaluating a model's prediction. In the COCO metric, mAP is calculated by taking the average of multiple IoU (Intersection over Union). mAP @0.5 means that the mAP is calculated by considering the IoU of 50% or more. Generally the threshold is set at 50% IoU. The COCO metric also calculates the mAP @0.5 to 0.95. That means the IoU range is between 50% and 95%.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

4.2.5 F1-Score

F1 score combines the results of precision and recall by calculating their harmonic mean. If a model produces higher precision value and another model produces higher recall value then a better model F1 score is used.

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{4}$$

4.3 Performance of the state of the art models

Table 2: Model Evaluation

Model	mAP @0.5	Average Precision	Average Recall	F1 Score
EfficientDet D0	0.661	0.661	0.454	0.5383
Faster RCNN ResNet 152	0.786	0.786	0.519	0.6252
Faster RCNN ResNet 101	0.799	0.799	0.532	0.6387
YOLOv5s	0.828	0.793	0.81	0.78
Faster RCNN Inception ResNet v2	0.839	0.839	0.556	0.6688
YOLOv5x	**0.921**	**0.9**	**0.899**	**0.90**

We evaluated six different state of the art object detection models with our dataset. From our result, YOLOv5x took the lowest converging time while EfficientDet D0 took the highest converging time. YOLOv5x got the highest mAP (Mean Average Precision) of 0.921 at 0.5 IoU threshold. On the other hand EfficientDet D0 model got the lowest mAP of 0.660802 at 0.5 IoU threshold. As EfficientDet is slow to converge, it required a high number of steps for training the dataset than the other models. Faster R-CNN Inception ResNet v2 took the highest training time per step among the 6 state of the art models. YOLOv5x model got the lowest training time per step among the models. As YOLO is a one shot object detection model, it performs faster than 2 shot object detectors such as EfficientDet and Faster R-CNN ResNet.

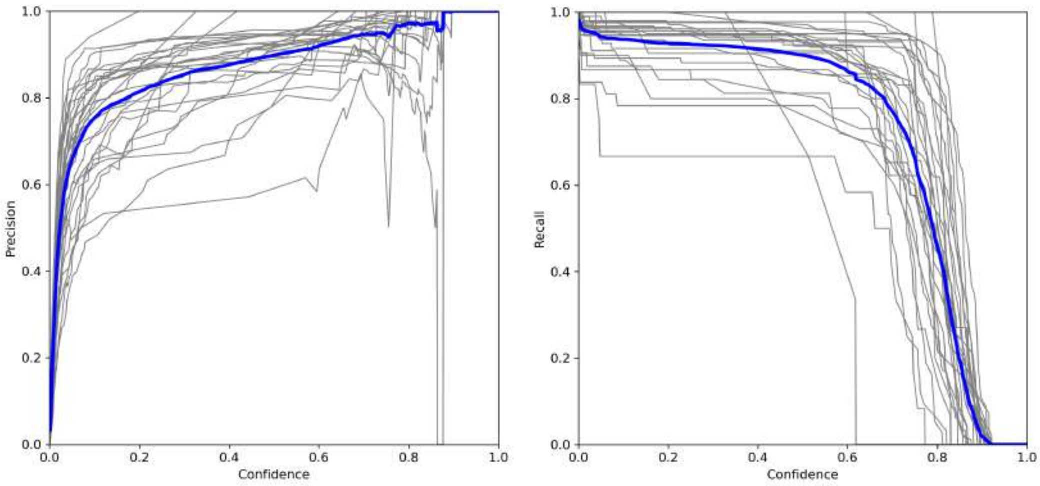


Figure 21: precision and recall curve on yolov5x

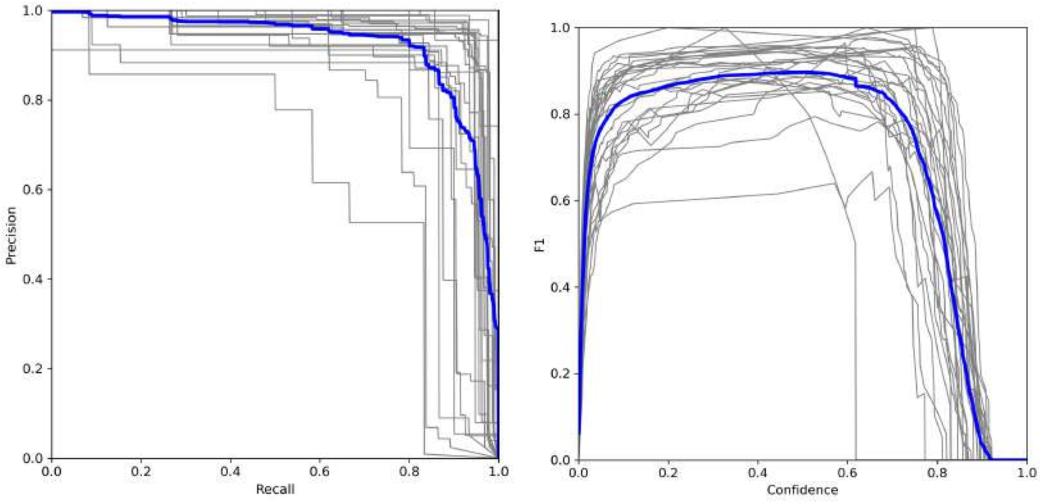


Figure 22: Precision-Recall and F1 curve on yolov5x

Figure 21 and 22 shows the precision, recall, precision-recall and f1 curve of each of the classes after evaluating the YOLOv5x model. The dark blue one is the aggregated curve for each of those classes. Here we can see that apart from a few classes, the curves got a smooth shape. In what confidence score the best value is achieved is also shown here. For example, for all classes, recall value 0 is achieved at the confidence score of 0.98.

4.4 Error Analysis

From all the six models we have experimented our dataset with, YOLOv5x has the least number of false positives. If we take a look at the confusion matrix of YOLOv5x from figure 23 we can identify the pair of classes where the model got confused to distinguish between them and made false predictions. Even in YOLOv5x we can see from the confusion matrix that the most confused pair of classes are 'Sharp Bend To The Right' and 'Sharp Bend To The Left', 'Cross Roads' and 'Staggered Junction', 'Double Bend First Left' and 'Double Bend First Right' from the intersection of those pairs.

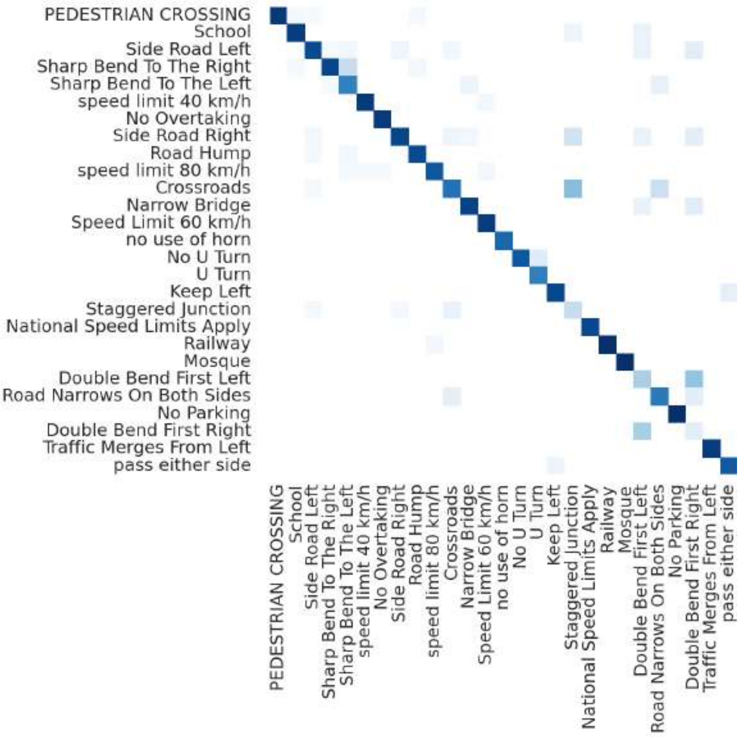


Figure 23: Confusion matrix of yolov5x

While analyzing the false positives in other models, we have also seen that apart from EfficientDet, all other models often got confused between 'Sharp Bend To The Right' and 'Sharp Bend To The Left', 'Cross Roads' and 'Staggered Junction', 'Double Bend First Left' and 'Double Bend First Right'.

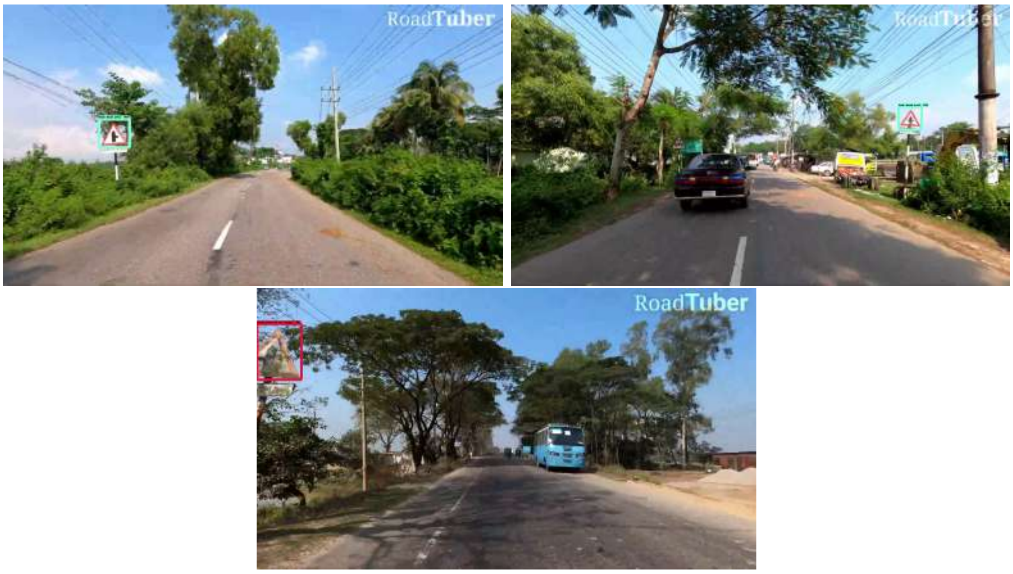


Figure 24: Example of similar types of signs

Figure 24 shows some samples of misclassified images by our trained models. Here the actual signs are 'Side Road Right', 'Crossroads', 'Side Road Left'. Those 3 signs have quite some similarities and all the Faser RCNN ResNet models failed to correctly identify them. On the other hand EfficientDet failed to even detect a lot of objects.

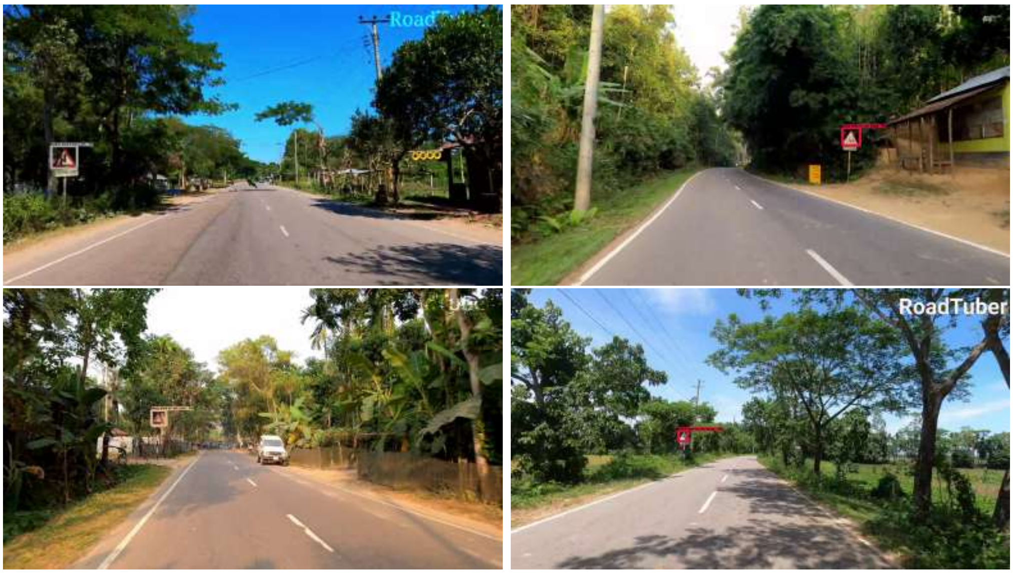


Figure 25: Example of similar types of signs

One if the most similar pairs of signs are ’Double Bend First Left’ and ’Double Bend First Right’. In figure 25 we can see two instances where ’Double Bend First Left’ is said ’Double Bend First Right’ and vice versa. In some cases we have also seen that ’Sharp Bend To The Right’ and ’Sharp Bend To The Left’. We have also found some other misclassified cases where the labeled one has hardly any resemblance with the actual one but it mostly happened when the sign is at distance from the scene or has some other challenge in it.

4.5 Performance of Models on Challenging Images

In figure 26, we can see, the traffic sign “No U Turn” is quite far from the camera. YOLOv5x detected the small traffic sign with 68% confidence score while YOLO v5s detected the small traffic sign with 55% confidence score.

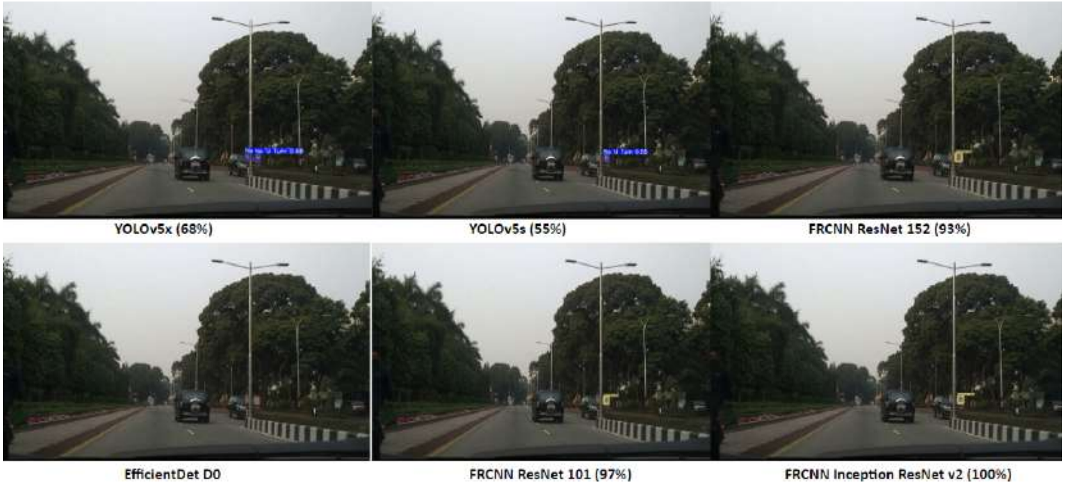


Figure 26: Comparison of Different Models (Small TS)

As YOLO is a one shot detection algorithm, it does not perform well on small object detection. That is why we can see that the confidence score is very smaller than the other models’ confidence scores, though it has the highest mAP value. Faster RCNN Inception ResNet v2 detected the small traffic sign with 100% confidence score. One probable reason might be that it took longer training time per step which helped it to learn the complex features of the objects efficiently and thus got the highest confidence score with small object. Unfortunately EfficientDet-D0 could not detect the traffic sign. As per our experiment, EfficientDet-D0 achieved the lowest performance than other models.

This is supposed to be the reason behind it's failure of detecting small objects. Faster R-CNN ResNet152 and Faster R-CNN ResNet101 detected the small object with 93% and 97% confidence score respectively which is supported by their mAP values.

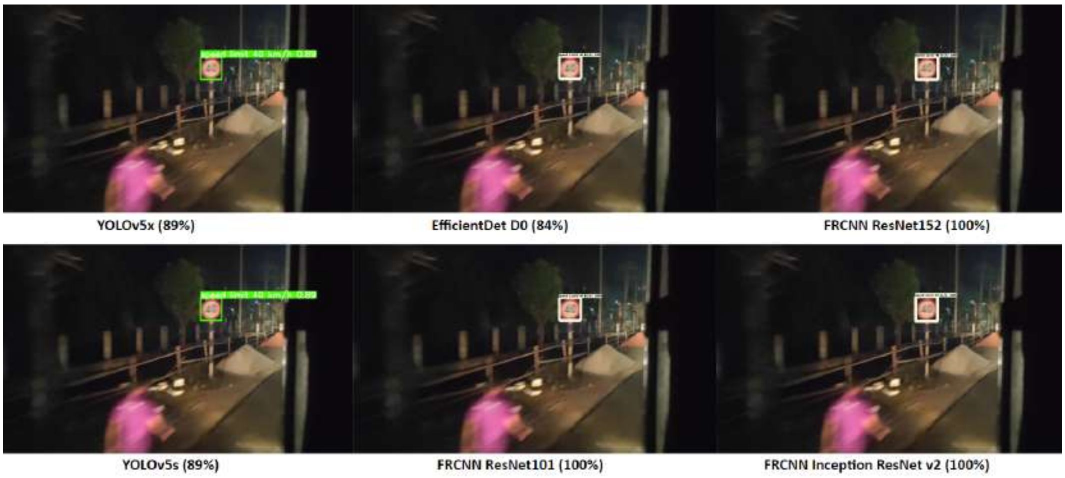

YOLOv5x (89%)
EfficientDet D0 (84%)
FRCNN ResNet152 (100%)
YOLOv5s (89%)
FRCNN ResNet101 (100%)
FRCNN Inception ResNet v2 (100%)

Figure 27: Comparison of Different Models (Night & Blurry TS)

From figure 27, the traffic sign "speed limit 40 km/h" is quite blurry. Also it is a night image. Faster RCNN ResNet 152, Faster RCNN ResNet 101 and Faster RCNN Inception ResNet v2, all the 3 models detected the blurry traffic sign of the night image with 100% confidence score. But the EfficientDet D0 model detected the small object with the lowest confidence score of 84%. From figure 12, EfficientDet D0 got the lowest mAP among other models which supports the lowest confidence score of EfficientDet D0. YOLOv5x and YOLO v5s both got 89% confidence score which is quite good according to their result from figure 12.

4.6 Performance of Models on Background Images

In this phase of our research we took our previously collected 2000 background images and passed them through each saved models. We wanted to see how the models work in terms of distinguishing between traffic signs and traffic sign like objects.

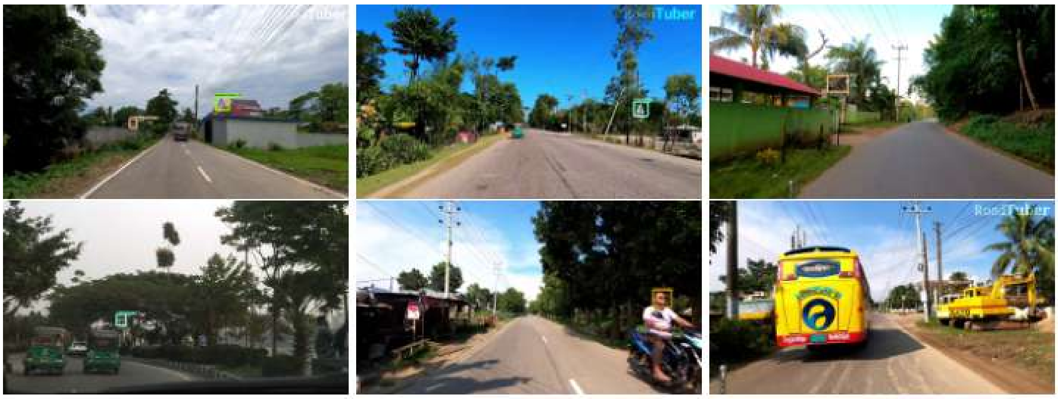


Figure 28: False detection in non traffic sign dataset

From figure 28, the first image from the top left corner is misclassified into 2 classes, “Sharp Bend to The Left” and “PEDESTRIAN CROSSING”. As “PEDESTRIAN CROSSING” is the class with the highest number of samples which is 975, there is a tendency to label objects as “PEDESTRIAN CROSSING”. The class “Sharp Bend to The Left” has 466 samples which is a decent amount. The models found a curved object in the image and thus wrongly labelled it as “Sharp Bend to The Left” class. The image on the right side of the first one has a traffic sign facing the opposite side. The model wrongly detected and labelled the object as “Side Road Left”. The next image on the right side again portrays the wrongly classified “Sharp Bend to The Left” class. Again we can find a curved object which is the reason for this misclassification. The image on the bottom left corner shows the detection of the class “Side Road Left” even though there is nothing but leaves and branches of a tree. The model found a pattern the seemed similar to the sign “Side Road Left” and thus wrongly classified it into “Side Road Left”. In the image next to the right of it, we can see that the human face with a glass is wrongly detected as the class “no use of horn”. The most probable reason for this misclassification might be the shape of the spectacles is similar to the shape of the horn sign of the “no use of horn” class. We have seen this tendency in other images also. Finally in the image to the right of it, we see that the round logo behind the bus is detected as “No Parking” class. In the “No Parking” traffic sign, there is a leaning line in the middle of the round sign. We can also find a curved line in the middle of the round logo. Thus the models wrongly classified it.

Table 3: False detection in non traffic sign dataset

Model	**Detected Non-TS images (out of 2000)**
EfficientDet D0	8
Faster RCNN Inception ResNet v2	35
Faster RCNN ResNet 101	35
YOLOv5s	39
Faster RCNN ResNet 152	43
YOLOv5x D0	48

Table 3 shows for each class the number of detected objects in the background images. Because of the overall performance of EfficientDet D0 was not good it has the least amount of misclassified images. The rest of the models did not vary much in terms of distinguishing traffic sign like objects from the real ones. For all of those models we can see that in less than 2.5% of the cases, the models incorrectly identified the objects from the scenes as traffic signs.

Chapter 5

Conclusion

5.1 Summary

A smart traffic sign detection system can go a long way in ensuring safe driving, reducing unwanted road accidents and pave the path for autonomous driving at large. In this regard, we have curate a benchmark dataset for effectively classifying different traffic signs available in the roads of Bangladesh. The dataset mimics most of the real world scenarios and contains traffic sign images under several challenging conditions such as blurriness, occlusion, different weather and lighting conditions, and so on. The samples have been collected from different geographical locations and from a wide variety of vehicles. After preparing the dataset we have used it to provide a baseline by evaluating six different state-of-the-art object detection models. From our experimental results we have observed that YOLOv5x model has achieved the best performance in terms of mAP, average precision and average recall. Furthermore, we have provided a thorough qualitative and quantitative analysis against the achieved results. Finally, we checked the performance of the models on a set of non-traffic images to justify the generalization capability and achieved satisfactory performance. However, we have observed that distinguishing between the traffic signs and traffic sign like objects is a major challenge to be tackled in this area of research.

5.2 Limitations

While curating the dataset, we selected only the classes having more or equal to 50 samples. This threshold can be increased to a higher value to reduce the amount of class imbalance of the dataset. The intra-class similarity of different traffic signs have often reduced the performance of the classifier. The number of images from different weather conditions is still not enough. Moreover, the model mistakenly identifies a few traffic sign like structures as traffic signs when it was tested with the background images.

5.3 Future Work

In future, the overlapping of different publicly available datasets can be exploited to enrich the dataset. More instances can be added to the classes with a lower number of samples to solve the class imbalance issue. The dataset can be further enriched by adding more challenging scenarios like diversified weather conditions, samples from different geographical locations and a wide variety of vehicles, increased background variations, etc. A good amount of work is yet to be done to effectively distinguish between the traffic sign images from the traffic sign like objects.

References

[1] A. Gautam and S. Singh, “Deep learning based object detection combined with internet of things for remote surveillance,” *Wireless Personal Communications*, vol. 118, no. 4, p. 2121–2140, 2021.

[2] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, “Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[3] R. Gavrilescu, C. Zet, C. Fosalau, M. Skoczylas, and D. Cotovanu, “Faster r-cnn:an approach to real-time object detection,” *2018 International Conference and Exposition on Electrical And Power Engineering (EPE)*, 2018.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] M. Tan, R. Pang, and Q. V. Le, “Efficientdet: Scalable and efficient object detection,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] R. Timofte, K. Zimmermann, and L. Van Gool, “Multi-view traffic sign detection, recognition, and 3d localisation,” *Machine vision and applications*, vol. 25, no. 3, pp. 633–647, 2014.

[7] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.

[8] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” 2019. [Online]. Available: https://arxiv.org/abs/1905.05055

[9] J. Li, D. Zhang, J. Zhang, J. Zhang, T. Li, Y. Xia, Q. Yan, and L. Xun, “Facial expression recognition with faster r-cnn,” *Procedia Computer Science*, vol.

107, pp. 135–140, 2017, advances in Information and Communication Technology: Proceedings of 7th International Congress of Information and Communication Technology (ICICT2017). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050917303447

[10] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742030614X

[11] Z. Xu, W. Yang, A. Meng, N. Lu, H. Huang, C. Ying, and L. Huang, "Towards end-to-end license plate detection and recognition: A large dataset and baseline," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 255–271.

[12] A. Arcos-Garcia, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Evaluation of deep neural networks for traffic sign detection systems," *Neurocomputing*, vol. 316, pp. 332–344, 2018.

[13] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, "Vulnerable objects detection for autonomous driving: A review," *Integration*, vol. 78, pp. 36–48, 2021.

[14] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[15] K. Xiao, L. Engstrom, A. Ilyas, and A. Madry, "Noise or signal: The role of image backgrounds in object recognition," *arXiv preprint arXiv:2006.09994*, 2020.

[16] S. B. Wali, M. A. Abdullah, M. A. Hannan, A. Hussain, S. A. Samad, P. J. Ker, and M. B. Mansor, "Vision-based traffic sign detection and recognition systems: Current trends and challenges," *Sensors*, vol. 19, no. 9, p. 2093, 2019.

[17] T. R. . November and T. Report, “Bangladesh 106th among 183 countries for having most road accidents: Report,” Tech. Rep., Nov 2021. [Online]. Available: https://www.tbsnews.net/bangladesh/bangladesh-106th-among-183-countries-having-most-road-accidents-report-335299

[18] K. Maniruzzaman and R. Mitra, “Road accidents in bangladesh,” *IATSS research*, vol. 29, no. 2, p. 71, 2005.

[19] “Road traffic injuries,” https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2021.

[20] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel, “Detection of traffic signs in real-world images: The german traffic sign detection benchmark,” in *The 2013 international joint conference on neural networks (IJCNN)*. Ieee, 2013, pp. 1–8.

[21] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “The german traffic sign recognition benchmark: a multi-class classification competition,” in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[22] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, “Traffic-sign detection and classification in the wild,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2110–2118.

[23] D. M. Ramík, C. Sabourin, R. Moreno, and K. Madani, “A machine learning based intelligent vision system for autonomous object detection and recognition,” *Applied intelligence*, vol. 40, no. 2, pp. 358–375, 2014.

[24] A. Vennelakanti, S. Shreya, R. Rajendran, D. Sarkar, D. Muddegowda, and P. Hanagal, “Traffic sign detection and recognition using a cnn ensemble,” in *2019 IEEE international conference on consumer electronics (ICCE)*. IEEE, 2019, pp. 1–4.

[25] A. Shustanov and P. Yakimov, “Cnn design for real-time traffic sign recognition,” *Procedia engineering*, vol. 201, pp. 718–725, 2017.

[26] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[27] L. Wu, H. Li, J. He, and X. Chen, "Traffic sign detection method based on faster r-cnn," in *Journal of Physics: Conference Series*, vol. 1176, no. 3. IOP Publishing, 2019, p. 032045.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[29] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[30] X. Changzhen, W. Cong, M. Weixin, and S. Yanmei, "A traffic sign detection algorithm based on deep convolutional neural network," in *2016 IEEE International Conference on Signal and Image Processing (ICSIP)*. IEEE, 2016, pp. 676–679.

[31] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[32] S. P. Rajendran, L. Shine, R. Pradeep, and S. Vijayaraghavan, "Real-time traffic sign recognition using yolov3 based detector," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–7.

[33] A. Ellahyani, M. Ansari, I. Jaafari, and S. Charfi, "Traffic sign detection and recognition using features combination and random forests," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 686–693, 2016.

[34] C. G. Serna and Y. Ruichek, "Classification of traffic signs: The european dataset," *IEEE Access*, vol. 6, pp. 78 136–78 148, 2018.

[35] X. Bangquan and W. X. Xiong, “Real-time embedded traffic sign recognition using efficient convolutional neural network,” *IEEE Access*, vol. 7, pp. 53 330–53 346, 2019.

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[37] A. Jose, H. Thodupunoori, and B. B. Nair, “A novel traffic sign recognition system combining viola–jones framework and deep learning,” in *Soft Computing and Signal Processing*. Springer, 2019, pp. 507–517.

[38] K. Kaplan, C. Kurtul, and H. L. Akin, “Real-time traffic sign detection and classification method for intelligent vehicles,” in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*. IEEE, 2012, pp. 448–453.

[39] J. Zhang, Z. Xie, J. Sun, X. Zou, and J. Wang, “A cascaded r-cnn with multiscale attention and imbalanced samples for traffic sign detection,” *IEEE Access*, vol. 8, pp. 29 742–29 754, 2020.

[40] J. Cao, C. Song, S. Peng, F. Xiao, and S. Song, “Improved traffic sign detection and recognition algorithm for intelligent vehicles,” *Sensors*, vol. 19, no. 18, p. 4021, 2019.

[41] Y. Jin, Y. Fu, W. Wang, J. Guo, C. Ren, and X. Xiang, “Multi-feature fusion and enhancement single shot detector for traffic sign recognition,” *IEEE Access*, vol. 8, pp. 38 931–38 940, 2020.

[42] H.-Y. Lin, C.-C. Chang, V. L. Tran, and J.-H. Shi, “Improved traffic sign recognition for in-car cameras,” *Journal of the Chinese Institute of Engineers*, vol. 43, no. 3, pp. 300–307, 2020.

[43] A. Avramović, D. Sluga, D. Tabernik, D. Skočaj, V. Stojnić, and N. Ilc, “Neural-network-based traffic sign detection and recognition in high-definition images using region focusing and parallelization,” *IEEE Access*, vol. 8, pp. 189 855–189 868, 2020.

[44] B. B. Fan and H. Yang, "Multi-scale traffic sign detection model with attention," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 235, no. 2-3, pp. 708–720, 2021.

[45] S. M. M. Ahsan, S. Das, S. Kumar, and Z. La Tasriba, "A detailed study on bangladeshi road sign detection and recognition," in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, 2019, pp. 1–6.

[46] S. Chakraborty, M. N. Uddin, and K. Deb, "Bangladeshi road sign recognition based on dtbs vector and artificial neural network," in *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2017, pp. 599–603.

[47] Y. Fan and W. Zhang, "Traffic sign detection and classification for advanced driver assistant systems," in *2015 12th international conference on Fuzzy systems and knowledge discovery (FSKD)*. IEEE, 2015, pp. 1335–1339.

[48] T. Bui-Minh, O. Ghita, P. F. Whelan, and T. Hoang, "A robust algorithm for detection and classification of traffic signs in video data," in *2012 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2012, pp. 108–113.

[49] M. Shahed, M. A. U. Khan, and S. A. Chowdhury, "Detection and recognition of bangladeshi road sign based on maximally stable extremal region," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*. IEEE, 2017, pp. 1–6.

[50] P. Dhar, M. Z. Abedin, T. Biswas, and A. Datta, "Traffic sign detection—a new approach and recognition using convolution neural network," in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*. IEEE, 2017, pp. 416–419.

[51] Y. Sun, P. Ge, and D. Liu, "Traffic sign detection and recognition based on convolutional neural network," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 2851–2854.

[52] *Bangladesh Road Sign Manual*, 2000, vol. 1 and 2. [Online]. Available: https://bsp.brta.gov.bd/roadSignMannul;jsessionid=mRcupj7oFff6rDG9QKvVOBturxwVFmDYUlMnD3dkPICYxuDMbQ5n!1875640356?lan=en

[53] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7310–7311.