Road Sign Detection Using Variants of YOLO and R-CNN: An Analysis from the Perspective of Bangladesh

Aklima Akter Lima, Md. Mohsin Kabir, Sujoy Chandra Das, Md. Nahid Hasan, and M. F. Mridha

Abstract Road sign detection represents a feature that assures the safety of drivers, vehicles, and pedestrians by efficiently detecting road signs. This feature is designed to notify drivers about road signs whether he is missing the signs or not. This detecting and recognizing feature of the road signs' has improved a part of the advanced driver assistance system (ADAS). ADAS is an automated technology containing cameras and sensors intended to help the drivers with road signs, while traveling to a new road or having no knowledge about road signs. Before the work analysis, this topic has shown formidability as it has a real-time processing solution. This paper analyzed seven architectures for detecting the road signs: YOLO, YOLOv2, YOLOv3, PP-YOLO model and R-CNN, Fast R-CNN, Faster R-CNN. We have built a dataset based on Bangladesh's road sign named the "BD Road Sign 2021 (BDRS 2021)" dataset to evaluate the architectures. This dataset contains 16 categories (16 types of road-sign), and each has 168 images. Finally, we applied the seven advanced architectures to find the effective one to detect Bangladesh's road signs. This study implies that YOLOv3 and Faster R-CNN perform comparatively better for road sign detection.

Keywords Advanced driver assistance system (ADAS) · Road sign detection · You only look once (YOLO) · Region-based convolutional neural networks (R-CNN) · Deep learning

1 Introduction

Identifying road sign's position is an important area of research that continuously captures the attention of researchers in the area of Intelligent Transportation System (ITS). The road sign shows route road marking, possible hazards, and involvements

A. A. Lima · Md. M. Kabir (✉) · S. C. Das · Md. N. Hasan · M. F. Mridha
Bangladesh University of Business and Technology, Dhaka, Bangladesh

M. F. Mridha
e-mail: firoz@bubt.edu.bd

M. S. Arefin et al. (eds.), *Proceedings of the International Conference on Big Data, IoT, and Machine Learning*, Lecture Notes on Data Engineering and Communications Technologies 95, https://doi.org/10.1007/978-981-16-6636-0_42

that vehicles can encounter on the road. Adding to that it assists vehicles in route by providing valuable data and alerts. Every driver must hold their attention on the road and be aware of their surroundings while driving as road signs have variations everywhere in the world. This is not yet possible to construct the universal Traffic Sign Recognition System (TSRS) structure. When a driver is introduced to a different route, he must concentrate on the road incredibly late at night, which results in a diversion from a highway sign. Road sign detection features could be a helpful way to assist drivers and mitigate road injuries caused by the driver's lack of understanding. A system or feature should be built for Bangladeshi road signs to alert drivers to road signs without interfering with their driving concentration. As a result, we are firmly persuaded to conduct some research in this direction with the explicit aim of providing more analysis in the study of TSRS concerning the Bangladesh environment.

The driver will be directed into a favorable configuration every time with support from an Advanced Driver Assistance System (ADAS) for any encountered signs. As a result, drivers won't have to face finding out the sign's meaning, and a better TSRS structure can make it possible. TSRS structure techniques are divided into two key sections: position and identification. The design helps to support the driver in several ways to ensure their well-being, also the safety of various people and pedestrians on the path. These systems include one main goal: to identify and track road signs mostly during the driving period. With these features, the system will direct and make drivers aware of the consequences to the environment. This article focuses on creating a TSRS framework for Bangladeshi road signs that use some architecture based segment measurement and identification. An empirical analysis and its empirical setup comparing the following seven main architectures: YOLO [1], YOLOv2 [2], YOLOv3 [3], PP-YOLO [4] model and R-CNN [5], Fast R-CNN [6], Faster R-CNN for detecting signs. The overall contributions of this research are:

- We have investigated and distinguished the contemporary challenges of the Advanced Driver Assistance System.
- A recently built dataset named "BD Road sign 2021 (BDRS 2021)" is introduced that consists of 16 classes. Each class consists of 168 images.
- We applied seven baseline architectures, "YOLO, YOLOv2, YOLOv3, PP-YOLO model and R-CNN, Fast R-CNN, Faster R-CNN", to the newly created dataset and analyzed the obtained results. The analysis found Faster R-CNN and YOLOv3 more effective.

The continuation of this experimental paper is organized as follows: The previous literature is described in Sect. 2. Section 3 describes the dataset. Section 4 addresses the process, including features such as data preprocessing and design. Section 5 explains the assessment, provides a summary of the experiments as well. Finally, Sect. 6 brings the article to a close.

2 Related Work

Due to technical advancements like computing and computer vision in the modern age, a device allows quick, accurate, and automatic detection of road signs in various conditions. Many notable cutting-edge architectures have been developed in recent years. Among those listed are:

A novel approach for traffic sign detection based on deep learning architecture named capsule networks achieves excellent performance on the German road sign dataset, which is introduced in [7]. In some cases, CNN's are easily fooled by multiple adversary attacks [8], but capsule networks can overcome those attacker attacks and improve traffic sign detection accuracy. Compared to CNNs, capsule networks perform much better by correctly performing image classification and recognition tasks [9]. D. Tabernik and D. Skočaj identified and recognized a wide range of traffic sign categories appropriate for automating traffic sign inventory management. The mask R-CNN is a CNN-based approach that addresses the entire detection and recognition process with automated end-to-end learning. This method is used to detect the 200 traffic sign classes specified in the dataset. Researchers demonstrated that the deep learning-based approach could produce an outstanding performance for a wide range of traffic sign categories, along with some complex ones with high intra-class variability [10].

Wang and Guo [11] suggested the YOLO neural network model is configured using an updated CNN model focused on the YOLO model, darknet 53. By adding batch normalization and RPN networks, it can enhance network architecture for traffic sign detection. The method described in this paper will significantly improve the efficiency and detection rate of traffic signs, while also reducing the detection system's hardware specifications. The authors of [12] use the Radial Symmetry Transform to identify other geometric shapes such as octagons, squares, and triangles.

Zhang, J. suggested an end-to-end convolutional network modeled after YOLOv2. To detect minor traffic signs more effectively, they divide the input images into dense grids and generate more precise feature maps. Both experimental results based on their extended CTSD and German Traffic Sign Detection Framework (GTSDB) show that the proposed approach is faster and more stable [13]. Buyval, presented a technique for classifying and localizing road signs in 3D space using a neural network and a point cloud acquired from a laser range finder (LIDAR). A dataset was collected to achieve this goal and train the neural network (built on the Faster-R-CNN architecture). The device generates a series of images with bounding boxes and points clouds related to actual road signs [14]. The first section of a method for detecting and classifying road signs identifies the road signs on a real-time basis. The second section identifies the German traffic signs (GTSRB) dataset and produces predictions using the road signs detected during the first section. In the detection section, they used HOG and SVM to identify the road signs captured images. Later, in the classification section, a convolutional layer based on the LeNet model was used to modify [15]. A system for detecting and recognizing Bangladeshi road signs is being established. To begin, images of road signs are collected from various districts

across Bangladesh to create the dataset. The photos are then numbered, and the Single Shot Multibox Detector (SSD) is then used to locate and identify road signs. A CNN-based model is being used in the classification stage [16].

Besides, a large number of deep learning-based road sign detection approaches are proposed in the last decades. Our paper mainly analyzes the road signs that are used in Bangladesh based on seven detection algorithms.

3 Dataset

We discovered that most road sign image datasets on the web are divided into four categories: regulatory signs, warning signs, information signs, and additional signs. We attempted to gather simple road signs commonly used on Bangladesh's roads from the set of every road sign. We have gathered a large number of images from the Bangladesh Road Transport Corporation (BRTC). Finally concludes, 168 images for each of the 16 types (Under four classes: regulatory signs, warning signs, information signs, and additional signs) and used 80% (2150) of the photos for training and the remaining 20% (538) for testing in our assessments (Fig. 1).

4 Methodology

To test road sign detection from image datasets "BDRS 2021," a comparative analysis of YOLO, YOLOv2, YOLOv3, PP-YOLO model, and R-CNN, Fast R-CNN, Faster R-CNN architectures is introduced and benchmarked. The framework is outlined in detail in the articles that follow.

4.1 Data Preprocessing

Data preprocessing is divided into two stages: data normalization and data augmentation. These two methods are discussed further below.

Data Normalization: Image normalization is an essential preprocessing technique. It decreases the inner-class function disparity and is regarded as intensity offsets. Since the intensity offsets are defined in the field distribution, standard deviation and Gaussian normalization can be used to normalize. Equation (1) is used to evaluate the image during normalization [17].

$$\Psi(\pi, \theta) = \frac{\xi(\pi, \theta) - \mu(\pi, \theta)}{6\sigma(\pi, \theta)} \tag{1}$$

Fig. 1 Sample images from BDRS 2021 dataset

where μ is a local mean and σ is a local standard deviation [1].

$$\mu(\pi, \theta) = \frac{1}{M^2} \sum_{k=-\alpha}^{\alpha} \sum_{n=-\alpha}^{\alpha} \xi(K + \mu, n + \theta)$$

Data Augmentation: The data augmentation techniques are applied to enlarge the dataset. We used five image appearance filters: Gaussian, disk, unsharp, average and motion, and six affine transformation matrices. This makes the dataset quantity 30 times of its actual size.

4.2 Baseline Architectures

Seven baseline architectures of deep learning-based detection algorithms are used to evaluate the dataset. This section briefly analyzes the seven algorithms. The general structure of the evaluation system presents in Fig. 2.

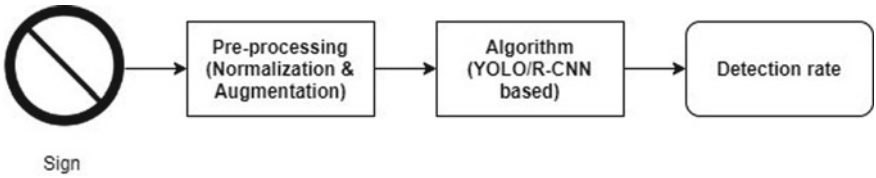

Fig. 2 The image presents the structure of the system. First, the photos were taken and passed through the preprocessing phase. Then each algorithm is applied to the processed data and results captured

YOLO: YOLO [1] is CNN-based for real-time object detection, which utilizes the entire image and splits it into regions, and estimates bounding boxes and probabilities for each image. The estimated probabilities are used to weight these bounding boxes when it reaches its high precision, while operating in real-time. The method is called "You Only Look Once" at the object in the context, which makes predictions. Then, it outputs objects predictions along with bounding boxes after non-max suppression. It generally learns applicable representations of objects, allowing it to perform better from many detection methods.

YOLOv2: YOLOv2 implements a range of enhancements to increase accuracy and batch processing. YOLOv2 [2] solves significantly higher localization error and poor recall comparison to region-based strategies by allowing batch standardization and better resolution classifiers. The Batch Normalization method is used to stabilize the input layers by modifying and measuring the activations [18]. Multi-scale instruction randomly selects a new size for every ten iterations of the system. This helps to predict well over a wide range of input measurements. The enhancement of the YOLOv2 is the really well functionality to enhance the ability to identify small items, which follow a pass-through layer method. This combines high-resolution features with low-resolution features, equivalent to ResNet identification mapping [19]. The mathematical analysis of the architecture can be found in [6].

YOLOv3: YOLOv3 [3] is published, distinguished by greater accuracy, and substitutes the softmax activation function with logistic regression and threshold. YOLOv3 is enhanced by using a multi-label classification that varies from the shared exclusive label used in the earlier versions. It utilizes a logistic classifier to measure the likelihood of the item becoming a particular mark. In classification loss, the binary cross-entropy loss by each mark is used rather than the generalized mean square error used in the earlier versions. The secondary enhancement is with a particular bounding box prediction that combines the score of one item in a bounding box anchor that overlaps the maximum likelihood object instead of others. YOLOv3 determines a bounding box anchor by each ground truth item. The third development with the use of estimation across dimensions by using the idea of feature pyramid networks. YOLOv3 forecasts boxes on three spatial dimensions, and then extracts the features from all those scales. The predicted outcome of the network is a 3D sensor that encodes the bounding box, item score, and class estimation. The fifth

upgrade is the latest CNN function extractor called Darknet-53. It's a 53-layer CNN that utilizes ResNet-inspired skip connections. YOLOv3 predicts at three different scales, precisely determined by downsampling the proportions of the source images by 32, 16, and 8 pixels, respectively.

PP-YOLO: The PP-YOLO [4] (PaddlePaddle YOLO) object detection system is based on the YOLO object detection algorithm. PP-YOLO is not a novel object detection system. Instead, PP-YOLO is a revised version of YOLOv4 with faster inference and a higher mAP score. Such enhancements are made possible by utilizing a RESNET-50 backbone architecture and additional features, including larger batch size, Drop block, IOU Loss, and training models. This structure consists of 3 parts, Backbone, Detection Neck, Detection Head. DarkNet-53 with ResNet50-vd, the backbone of YOLOv3, is substituted in PP-YOLO. It replaces some of the convolutional layers in ResNet50-vd with deformable convolutional layers (DCN) in this case. Many detection models have shown the efficacy of Deformable Convolutional Networks (DCN). ResNet as a backbone network architecture itself provided an increase in effectiveness and efficiency.

R-CNN: The region-based Convolutional Network (R-CNN) [5] method achieved excellent image prediction performance through using deep ConvNet to identify input images. The R-CNN [20] process trains CNN's end-to-end to locate the region proposals through element clusters or backgrounds. R-CNN development is a multi-stage process that includes extracting features, fine-tuning a log loss infrastructure, training SVMs, and eventually constructing a bounding box. There are drawbacks like R-CNN is sluggish since it executes the forward ConvNet transfer by each object proposal without exchanging the calculation and cannot modify the co-evolutionary layers that precipitate the pooling of the structural pyramid [21].

Fast R-CNN: The Fast R-CNN [22] network uses image data and a collection of training samples as input. The network processes the entire image with many co-evolutionary and max pooling to generate a fully connected function map. A region of interest (RoI) pooling layer extracts an adjusted vector function for each model for evaluation. Fast R-CNN has many advantages, such as higher recognition performance (mAP) than R-CNN, SPPnet, separate training, multi-task failure, and training can upgrade all network layers, and no storage devices are needed for caching features. In Fast R-CNN, we reduce an optimization technique after the multi-task loss.

Faster R-CNN: The Faster R-CNN [6] model consists of two components: the Region Proposed Network (RPN) and the Fast R-CNN tracker. RPN is an entirely convoluted system used to generate regional proposals with various dimensions and rotational speeds that serve as feedback for the second method. The RPN, as well as the Fast R-CNN detector, share a specific convolutional layer. Faster R-CNN, by extension, may be composed of a single and coordinated R-CNN. Network for the identification of artifacts. The RPN is a region proposal algorithm, and the Fast R-CNN as a detection network comprises the Faster R-CNN architecture.

5 Evaluation

The supervision of the empirical analysis on estimating the recommended Road sign detection on our "BDRS 2021" dataset carried out the comparative study of seven architectures. First, we describe the data set that was used in the research. Then, we describe the experimental setup. Third, the measures used to assess method accuracy are discussed. Fourth, the comparative study of seven architectures YOLO, YOLOv2, YOLOv3, PP-YOLO model and R-CNN, Fast R-CNN, Faster R-CNN are analyzed. Finally, the results of the comparative analysis are shown.

5.1 Evaluation Metric

Precision and recall measurement metrics are used to evaluate the architecture based on the confusion matrix results. Also, mAP is used, which is primarily used for the evaluation of visual object detectors.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^{k=n} \text{AP}_k \tag{4}$$

where TP means true positive, FP means false positive; FN means false-negative and AP_k is the AP of class k, and n is the number of classes. AP of class k calculates by the following formula.

$$\text{AP} = \sum_{k=0}^{k=n-1} [\text{Recalls}(k) - \text{Recalls}(k+1)] * \text{Precisions}(k) \tag{5}$$

where n is the number of thresholds.

5.2 Experimental Setup

Python is used for data preprocessing, experimentation, and model evaluation in the "BDRS 2021" dataset. TensorFlow [23] and Keras [24] are used to evaluate the proposed architecture. Furthermore, NumPy [25] is used to perform mathematical operations on seven architectures that are compared in our dataset for the experiment.

5.3 *Experiments and Comparisons*

In this paper, the "BDRS 2021" dataset is used to detect Bangladesh's road signs. The experiment is done based on the mentioned seven architectures of YOLO and R-CNN. For the YOLO algorithm, we applied the Darknet implementation. The Darknet implementation of YOLO gives 0.642 mAP for the proposed dataset. As the YOLO algorithm makes many localization errors and lower recall rates, the accuracy obtained is insufficient. The YOLOv2 further reduced these problems. This architecture is developed using Darknet-19 deep architecture and increases the mAP to 0.76 for the mentioned dataset. The faster YOLO version till present days is YOLOv3. We applied the Darknet-53 as the backbone architecture of YOLOv3 and obtained massive enhancements of the result to 0.885 mAP. Afterward, the dataset is also used for the PP-YOLO variant. PP-YOLO replaced the Darknet-53 backbone with ResNet architecture and became helpful for real application scenarios. We achieved nearly the same mAP as YOLOv3 of 0.878 for PP-YOLO. Hence, the study shows that YOLOv3 with Darknet-53 backbone and ResNet backbone both give satisfactory road sign detection results.

Then, the dataset is evaluated using R-CNN, Fast R-CNN, and Faster R-CNN architecture. First, the R-CNN architecture is applied with a selective search algorithm. It takes a considerable training time, and the mAP score is only 0.683, which is much lower than any YOLO architecture. Then, the Fast R-CNN architecture is applied, which is the advanced version of R-CNN. In this time, the training time drastically reduced, and the mAP score increases to 0.795. Finally, we applied Faster R-CNN-based on the region proposal network and found satisfactory performances of 0.896 mAP. Table 1 presents the precision, recall, and mAP scores of the mentioned algorithms.

However, the study suggests YOLOv3 and Faster R-CNN for Bangladeshi Road Sign detection.

Table 1 This table presents the precision, recall, and mAP score of different sign detection architectures

Model	Precision	Recall	mAP
YOLO [1]	0.597	0.625	0.642
YOLOv2 [2]	0.732	0.751	0.760
YOLOv3 [3]	**0.882**	**0.899**	**0.885**
PP-YOLO [4]	0.878	0.877	0.878
R-CNN [5]	0.696	0.656	0.683
Fast R-CNN [22]	0.783	0.796	0.795
Faster R-CNN [6]	**0.884**	**0.901**	**0.896**

6 Conclusion

This paper represents a comparative analysis of road sign detection techniques implemented on the "BDRS 2021" dataset. It is collected from a public source and modified later according to preferences. Various architectures, specifically YOLO, YOLOv2, YOLOv3, PP-YOLO, R-CNN, Fast R-CNN were practiced for methodological modification on this dataset. Among the mentioned architectures YOLOv3 and Faster R-CNN perform better detecting the road sign more precisely on our respective datasets. The implementation of this paper shows good possibilities to recognize a road sign and reduce the risk of an accident caused by disregarding the signs by drivers. We discovered that no comparative study on this topic was conducted focusing on Bangladesh's Roads during our analysis. That is why, we think this research can enhance the factors and possibilities for the researchers, while working on this topic in future.

Acknowledgements We thankfully acknowledge the assistance of the Advanced Machine Learning lab for their resource sharing and supports.

References

1. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
2. Jo K, Im J, Kim J, Kim DS (2017) A real-time multi-class multi-object tracker using YOLOv2. In: 2017 IEEE International conference on signal and image processing applications (ICSIPA). IEEE, pp 507–511
3. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. arXiv preprint arXiv:1804.02767
4. Long X, Deng K, Wang G, Zhang Y, Dang Q, Gao Y et al (2020) PP-YOLO: an effective and efficient implementation of object detector. arXiv preprint arXiv:2007.12099
5. Benjdira B, Khursheed T, Koubaa A, Ammar A, Ouni K (2019) Car detection using unmanned aerial vehicles: comparison between faster R-CNN and Yolov3. In: 2019 1st International conference on unmanned vehicle systems-Oman (UVS). IEEE, pp 1–6
6. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. arXiv preprint arXiv:1506.01497
7. Hinton GE, Sabour S, Frosst N (2018) Matrix capsules with EM routing. In: International conference on learning representations
8. Su J, Vargas DV, Sakurai K (2019) One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 23(5):828–841
9. Kumar AD (2018) Novel deep learning model for traffic sign detection using capsule networks. arXiv preprint arXiv:1805.04424
10. Tabernik D, Skočaj D (2019) Deep learning for large-scale traffic-sign detection and recognition. IEEE Trans Intell Transp Syst 21(4):1427–1440
11. Wang Z, Guo H (2019) Research on traffic sign detection based on convolutional neural network. In: Proceedings of the 12th international symposium on visual information communication and interaction, pp 1–5

12. Gudigar A, Jagadale BN, Mahesh PK, Raghavendra U (2012) Kernel based automatic traffic sign detection and recognition using SVM. In: International conference on eco-friendly computing and communication systems. Springer, Berlin, Heidelberg, pp 153–161
13. Zhang J, Huang M, Jin X, Li X (2017) A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. Algorithms 10(4):127
14. Buyval A, Gabdullin A, Lyubimov M (2019) Road sign detection and localization based on camera and Lidar data. In: Eleventh international conference on machine vision (ICMV 2018), vol 11041. International Society for Optics and Photonics, p 1104125
15. Bouti A, Mahraz MA, Riffi J, Tairi H (2019) A robust system for road sign detection and classification using LeNet architecture based on convolutional neural network. Soft Comput 1–13
16. Ahsan SMM, Das S, Kumar S, La Tasriba Z (2019) A detailed study on Bangladeshi road sign detection and recognition. In: 2019 4th International conference on electrical information and communication technology (EICT). IEEE, pp 1–6
17. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. PMLR, pp 448–456
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
19. Maldonado-Bascón S, Lafuente-Arroyo S, Gil-Jimenez P, Gómez-Moreno H, López-Ferreras F (2007) Road-sign detection and recognition based on support vector machines. IEEE Trans Intell Transp Syst 8(2):264–278
20. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
21. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916
22. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
23. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J et al (2016) Tensorflow: a system for large-scale machine learning. In: 12th (USENIX) symposium on operating systems design and implementation (OSDI'16), pp 265–283
24. Gulli A, Pal S (2017) Deep learning with Keras. Packt Publishing Ltd.
25. Oliphant TE (2006) A guide to NumPy, vol 1. Trelgol Publishing, USA, p 85