TRAFFIC SIGN DETECTION USING YOLOv8

Rahul Kumar[1], Aniket Gupta[1], Rajeswari D[2*]
Department of Data Science and Business Systems,
School of computing, College of Engineering, and technology,
SRM Institute of Science and Technology, Kattankulathur– 603203,
Chennai, India.
E-mail: rk0361@srmist.edu.in, ag4718@srmist.edu.in, drajiit@gmail.com
*Corresponding Author

Abstract: Traffic signals and other signs like parking, stop signs, etc have become very crucial in autonomous and self-driving cars as it helps the smart system to comply with the basic traffic rules along with that it helps navigate routes based on the signs thus enabling a more secure driving experience for the drivers. There have been a lot of new algorithms that have emerged in the past recent years regarding this. In this paper, we have used the new yolov8 object detection system to help us detect traffic signs as it is much faster and more precise than its previous iterations. To improve the algorithm, we have used a dataset comprising photos of traffic signs taken at different angles and different light intensities. This system can predict the traffic signs with 93% accuracy.

Keywords: Traffic sign detection; Deep Learning; yolov8; Autonomous vehicle

1. INTRODUCTION

There has been continuous development in the road infrastructure and connectivity all over the world. With the increase in the ease of connectivity, the number of vehicles on the roads has also been increasing steadily due to which the amount of congestion and other road-related safety incidents is increasing over the past few years. There are rules and regulations in place to tackle those issues like traffic signs, traffic police etc.

Traffic signs are crucial to maintain safety and coordination on the roads while driving. There are a lot of traffic signs, and each sign represents something that is significantly different from the others for example traffic lights, parking signs, stop and no entry signs to zebra crossing signs each sign has a very important part to play in the overall road safety. Previously it was just about teaching and training the drivers about the signs so that they could follow those signs and maintain the regulations to ensure smooth driving.

But with the increasing growth in the number of automated vehicles with auto drive, autopilot, and driver assistance in the market, training only the drivers are no the only method through which safety can be ensured.[1] The system should now also be equipped with models to detect the signs quickly and accurately in real-time. Currently, there is a lot of safety-related problems in these visualization and detection systems which require continuous research [2].

This research continuously focuses on the various functionality of a model for detection and depending on the functionality, traffic signals can be categorized into various categories. In each category, there are several subcategories with similar shapes and sizes but different purposes. This thus gives us an idea that traffic sign detection is actually a two-step process with detection being step one and classification step two. The detection step helps to identify the binding box of the sign and the classification step assigns it to its respective category if present.

In the recent years, the use of deep learning models and CNN has been increasing it helps in enhancing and improving the process of object detection models to tackle this problem in real time as CNNs are capable of extracting image features and classifying images with a great level of precision and accuracy.

Along with traffic sign detection deep learning algorithms nowadays can also be used in efficient route mapping, traffic accident detection [3], and driver misbehavior [4] among many others.

These models have been useful in collecting real-time information about the road and traffic to find road accidents[5] monitor speed limits, traffic jams,driver behaviour[6] etc. But compared to complex models lightweight traffic detection are considerably much useful to ensure safety.

This paper is based on the YoloV8 algorithm which is a new and improved version of the YOLO Family. This algorithm aims to develop an efficient real-time traffic sign detection model using a publicly available dataset.

Some of our contributions are:

- Usage of the latest yolov8 model to generate more accurate results.
- The time for detection reduces significantly with the use of yolov8 as this model is faster than its predecessor.

2. LITERATURE

Traffic sign detection has been around for years using different algorithms and tools to help for identification and detection of traffic road signs.

Around two decades ago traffic sign detection used to be dependent on manual engineering of features due to the lack of more modern and sophisticated models.

The prominent methods of those times include the Viola-Jones Detector [7] and HOG (Histogram of Gradients)[8]. Viola-Jones Detector were well-known real-time face detectors but their application was limited to only specific objects whereas HOG was useful in pedestrian detection, but it encountered challenges in scaling it to a wider range of traffic signs.

Although these methods were pivotal and innovative their restrictions like reliance on handcrafted features, and limited scalability made it difficult for real-life use.

The evolution towards deep learning revolutionized traffic sign detection as it helped in overcoming the limitations faced by traditional approaches. The use of methods like R-CNN [9] in 2014 introduced the concept of using selective search for object proposals and CNNs for feature extraction. Although they were effective, it was slow in detecting the objects due to the redundant feature computation and fixed algorithms.

Advancements like Fast RCNN [10] and SPPNet [11] improved the speed by using the entire image for extracting the feature Compared with R-CNN, SPPNet processes the image at conv layers only once time while R-CNN processes the image at conv layers as many times as there are region proposals.

However, its training still had to be done in multiple stages and SPPNet only fine-tunes its layers that are connected fully while simply ignoring the other layers.

The major upgrade came in 2015 with the introduction of Faster RCNN, which used a region proposal network and scraped the use of selective search. This helped in achieving near real-time object detection by letting the network learn region proposals. However, redundancy in subsequent detection stages still was an area of concern.

In 2017, Feature Pyramid Networks was proposed. If we dig into Faster RCNN, we see that it is mostly unable to catch small objects in the image. To solve this a simple image pyramid can be used to scale the image to different sizes and send it to the network.

All the above mentioned algorithm only detects objects for a local region within the image and doesn't work on the full image.

YOLO[12] algorithms belong to a one-step detection algorithm that uses a Convolutional Neural Network (CNN) by converting the detection problem into a regression problem.

The graph obtained is cut into square n*n size region and features are extracted from it. Features of all the various kinds of labels are learned using various data. The region from which the data is to be collected its horizontal and vertical target region is identified and all the necessary parameters are identified to predict the object.

3. PROPOSED METHODOLOGY:

3.1 Dataset description

The dataset includes some of the most commonly used road traffic signs that we see on a daily basis. The dataset includes images of all three traffic light signals it also contains images for parking signals, stop signals, zebra crossing, u turn sign, and others.

The dataset has a total of 21 different traffic sign categories in the split of 80/20 which includes 1376 training images and 229 testing images

3.2 Proposed Architecture

For the proposed model the dataset into two parts one is for training purposes and the other one is for testing.

From the training dataset, all the required features are extracted, and the model is then trained using the yolov8 algorithm. Once trained the data is then adjusted and filtered to remove some inaccuracy and then we test it with the testing dataset to get the output. Figure 1 explains the entire architecture of the proposed architecture.

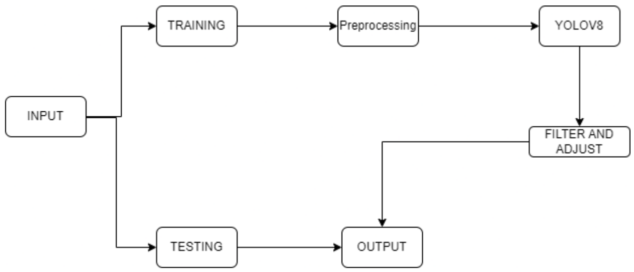


Fig 1 – Proposed architecture

3.2.1 Preprocessing

Preprocessing involves image resizing where the image are adjusted to the standard size, normalizing the pixel values for consistency, and annotating the bounding box and class labels for the image.

3.2.2 YOLOv8

Yolov8 consists of a Convolutional neural network (CNN) that breaks the image into a grid and identifies the bounding boxes and class for objects within all the grid cells at a time.

It uses an anchor box to improve accuracy for images of different sized and shapes and incorporate various methods like pyramid networks to enhance the performance .Figure 2 describes the general yolov8 architecture

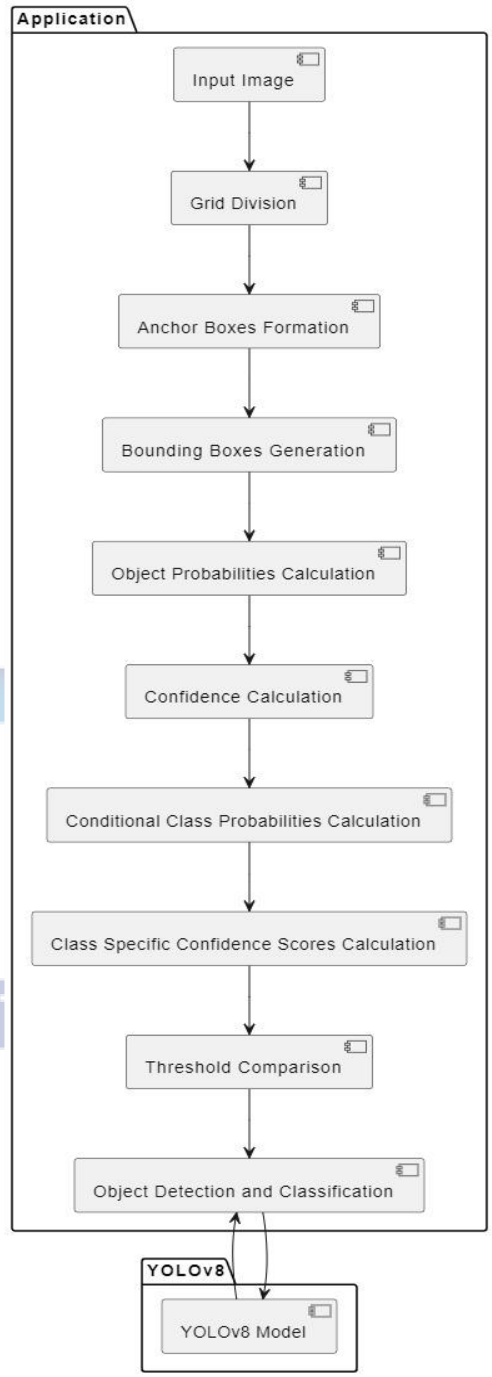


Fig 2 – yolov8 Model

4. RESULTS AND DISCUSSIONS

The dataset has a total of 21 different traffic sign categories in the split of 80/20 which includes 1376 training images and 229 testing images

The YOLO model has been trained on a device with intel core i5-10300H CPU, 8GB RAM, Nvidia GTX1650 GPU

4.1 PERFORMANCE EVALUATION

For detecting the accuracy of a yolov8 precision, accuracy , recall are calculated respectively in the given equation 1, 2 and 3.

$$\frac{TP}{TP+FP} \quad (1)$$

$$\frac{TP+TN}{TP+TN+FN+FP} \quad (2)$$

$$\frac{TP}{TP+FN} \quad (3)$$

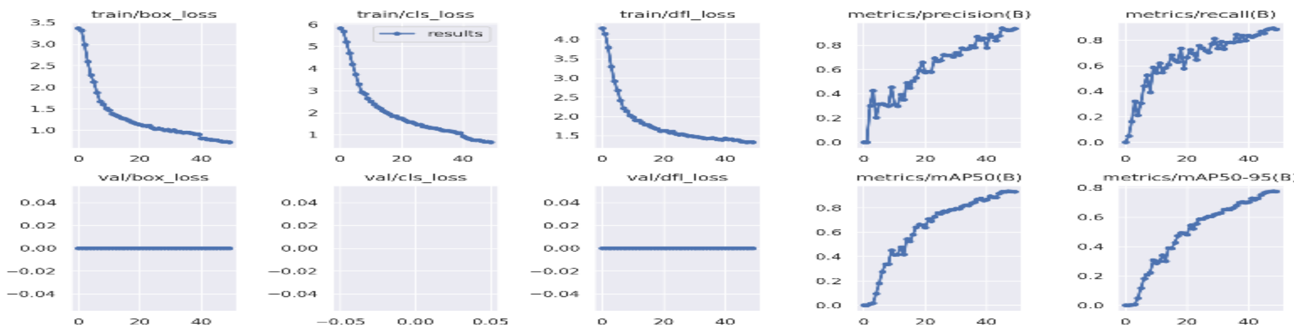
4.2 RESULTS

The dataset is first trained using yolov8 and after obtaining the details it is trained with yolov5 to find the difference in the mAP and check for any improvement between the 2 models.

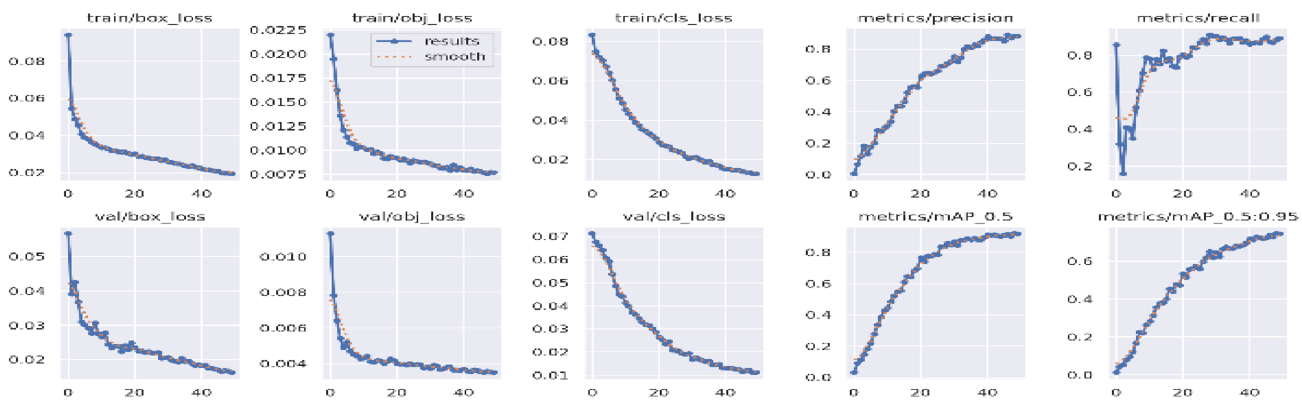
After training both the models the overall accuracy of yoloV8 was 0.935 whereas the overall accuracy of yoloV5 was 0.921.

Table 1. Comparision between YOLOv5 and YOOv8

	yoloV8			yoloV5		
class	mAP	precision	Recall	mAP	Precision	recall
all	0.935	0.936	0.885	0.921	0.883	0.881
do_not_enter	0.977	0.935	0.833	0.982	0.973	0.967
do_not_stop	0.955	0.944	0.867	0.946	0.807	0.9
do_not_turn	0.847	0.835	0.806	0.749	0.714	0.806
do_not_u_turn	0.962	0.966	0.8	0.944	0.908	0.655
enter_left_lane	0.97	0.937	0.998	0.972	0.938	1
Green_light	0.938	0.935	0.924	0.916	0.885	0.957
no_parking	0.884	0.903	0.824	0.666	0.562	0.588



Fig(2) – YoloV8 training graphs



Fig(3) – Yolov5 training graphs

Figure 2 represents the values of the data generated after training the dataset using yolov5 and Fig 3 represents the values of the data generated after training the same dataset using the yolov8 model.

Table 1 showcases the comparison between the accuracy, precision and recall of the training done with yolov8 and yolov5 respectively.

5. CONCLUSION

The YOLOv8 algorithm is utilized in this investigation for the purpose of detecting and classifying traffic signs. The scrutiny of experimental outcomes suggests that the credibility of traffic sign recognition is determined by the size of the dataset within the same algorithmic model, specifically YOLOv8. Enhanced classification and recognition rates are achieved with a larger dataset and optimal parameter adjustments.

This paper shows that by fine-tuning the model and parameters with the same data set, the model's performance and stability can be enhanced, and traffic sign recognition can be improved.

This paper also demonstrates that small data sets can be used for lightweight traffic sign classification models.

Future work may include A more accurate model with more categories of traffic sign, a more suited dataset with Indian road signs.

And improving the model for a better integration with a automated vehicle

6. REFERENCES

1. Rohit, R.V.S., Chandrawat, D., Rajeswari, D. "Smart Farming Techniques for New Farmers Using ML:, LNNS, 2021, vol 177.

2. Malik, S., Khan, M. A., & El-Sayed, H. (2021). Collaborative autonomous driving—A survey of solution approaches and future challenges. *Sensors*, *21*(11), 3783.

3. Sharma, S., Harit, S., & Kaur, J. (2022). Traffic Accident Detection Using Machine Learning Algorithms. *SUSCOM 2021* (pp. 501-507).

4 Kashevnik, A., Shchedrin, R., Kaiser, C., & Stocker, A. (2021). Driver distraction detection methods: A literature review and framework. *IEEE Access*, *9*, 60063-60076.

5. R. Mathur, T. Chintala and D. Rajeswari, "Detecting Criminal Activities and Promoting Safety Using Deep Learning," *ACCAI,* 2022, pp. 1-8.

6. D. Sikka, Shivansh, R. D and P. M, "Prediction of Delamination Size in Composite Material Using Machine Learning," *ICEARS,* 2022, pp. 1228-1232.

7. P. Viola and M. Jones, "Robust real-time face detection," *Proceedings Eighth IEEE International Conference on Computer Vision*. ICCV 2001, Vancouver, BC, Canada, 2001, pp. 747-747.

8. O. T. Khan and D. Rajeswari, "Brain Tumor detection Using Machine Learning and Deep Learning Approaches," *ACCAI*, 2022, pp. 1-7,

9. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 580-587.

10. R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV),* Santiago, Chile, 2015, pp. 1440-1448.

11. K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015.

12. 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),* Las Vegas, NV, USA, 2016, pp. 779-788.