Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 7, 2025

AUTONOMOUS ROAD SIGN DETECTION USING DEEP LEARNING MODELS

Muhammad Sajjad[*1], Aqeel Ahmad[2], Ayesha Batool[3], Muhammad Mudassar Naveed[4], Arslan Ejaz[5], Usman Aftab Butt[6]

[1,2,3,4,5]*Department of Computer Sciences, Minhaj University Lahore, Punjab Pakistan*
[6]*Department of Data Sciences, Minhaj University Lahore, Punjab Pakistan*

[*1]engineersajjad07cs21@gmail.com

DOI: https://doi.org/10.5281/zenodo.15920442

Keywords
Deep Learning Model, YOLOv11, IoT input unit, Road-sign

Article History
Received: 09 April, 2025
Accepted: 30 June, 2025
Published: 15 July, 2025

Corresponding Author: *
Muhammad Sajjad

Abstract
Detection of the object is an important technique for tracing and classifying objects in a picture or video. It is attained by sketching bounding boxes around the objects and allocating them consistent category labels. Earlier approaches, such as image filtering and training models on unambiguous weather situations, have proven insufficient. To overcome these problems, we propose a Deep Learning Model using YOLOv11 for robust traffic sign detection and identification. YOLOv11 suggests improvements in real-time object detection, surrounding tasks like instance segmentation, classification, and oriented object detection. The proposed model comprises an IoT input unit for image capture, a detection unit utilizing YOLOv11 for feature extraction and processing, and a voice activity detection unit to provide drivers with auditory alerts regarding detected signs. This integrated system aims to increase driving safety and efficiency by permitting vigorous and consistent traffic sign detection in real-world driving scenarios. The system is very efficient, achieving a weighted F1-score of 98.29%, weighted precision of 98.93%, weighted recall of 97.69%, weighted validation accuracy of 95.97% and macro validation accuracy of 96.17%.

INTRODUCTION

Detection of objects is an important technique for tracing and classifying objects in a picture or video [1]. Through the fast improvement of aided and autonomous driving technologies, the traffic sign detection system (TSDS) has become vital for civilizing driving safety and traffic productivity [2]. However, traffic signs are small in size in pictures, making them hard to detect [3]. The lighting and similar-looking objects also affect the detection of the road sign. The traffic sign detection system (TSDS) needs to work better in bad weather or low light. More research is needed to make working in dangerous situations. In the past, traffic sign detection used color and shape for the identification of the sign. Now, deep learning is being used for better results [4][5]. While traffic sign detection has improved, little focus should be on its performance in bad weather. Image filtering is being used to handle fog, low light, and rain in some studies, but this is slow and was not very good in real use [6] [7].

Some studies avoided image preprocessing and trained models on bad climate (fog, rain, and night) because of a lack of generalization; these models did not work properly on real roads because they lacked [8] [9]. Some research used the same traffic signs in adjacent pictures to improve precision. These methods worked well, but there was a problem with that because vehicles passed away quickly, and time

was wasted in recognizing the signal. This made it less useful in real driving [10]. It is vital to expand traffic sign detection in one go for real-world use.

In this study, we are proposing a Deep Learning Model (YOLOv11) for detecting and identifying the traffic signs in severe weather situations. YOLOv11 arises as a dominant and multipurpose improvement in the jurisdiction of real-time applications in the YOLO series. It can handle the various tasks, including instance segmentation, classification, position approximation, and orientation of objects [11]. YOLOv11 is constructed to monitor rules. It works in diverse circumstances. It can run with limited power on small devices. It also originates in various sizes to fit different needs. Finally, YOLOv11's improved performance and flexibility position. It's an appreciated tool for a varied range of applications [12].

Figure 1 Road-signs:

Related Works:

Detection algorithms involve One-stage detectors that detect objects in one go without extra processing [13], while Two-stage detectors first improve the accuracy by processing the detected objects and then handle classification. Two-stage detectors involve R-CNN, Fast R-CNN, and Faster R-CNN [14] [15][16]. In 2014, R-CNN was presented by Girshick and his team. This model was used for detecting and locating objects in pictures, by gaining high accuracy. In 2017, the Region Proposal Network (RPN) was introduced by Girshick and his team [17]. It public image with the detection network, forming region suggestions with nearly no additional cost and improves the accuracy [18]. One-stage detectors include SSD, RetinaNet, and YOLO [19] [20]. Redmon and his team introduced YOLO in 2016 [19]. It identifies objects by detecting their locations and classes in one go. This made detection fast and effective for real-time applications. YOLO series is becoming most famous due to speed, efficiency, and accuracy, and day by day improvements in architectures for real-time applications [21]. The timeline evolution of the YOLO series is given in Figure 1.

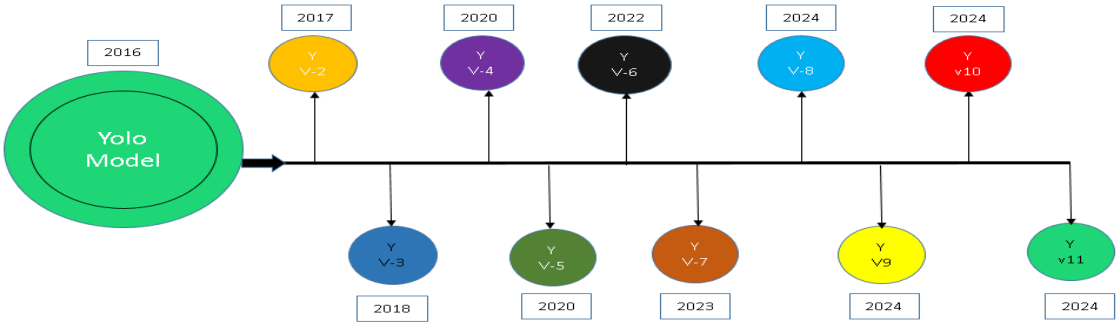

Figure 2-YOLO Evolution Timeline

We are proposing a model in which we use YOLOv11 for object detection. YOLOv11 is the most recent model in the YOLO series, having great speed, accuracy, and efficiency for object detection, even though they are very small. YOLOv11 arises as a dominant and multipurpose improvement in the jurisdiction of real-time applications in the YOLO series. It can handle various tasks, instance segmentation, classification, position approximation, and orientation of the object. YOLOv11 is constructed to monitor rules. It works in diverse circumstances. It can run with limited power on small devices. It also originates in various sizes to fit different needs [22]. The YOLOv11 model architecture is given below in Figure 2 [11].

S Nikhileswara Rao et al.

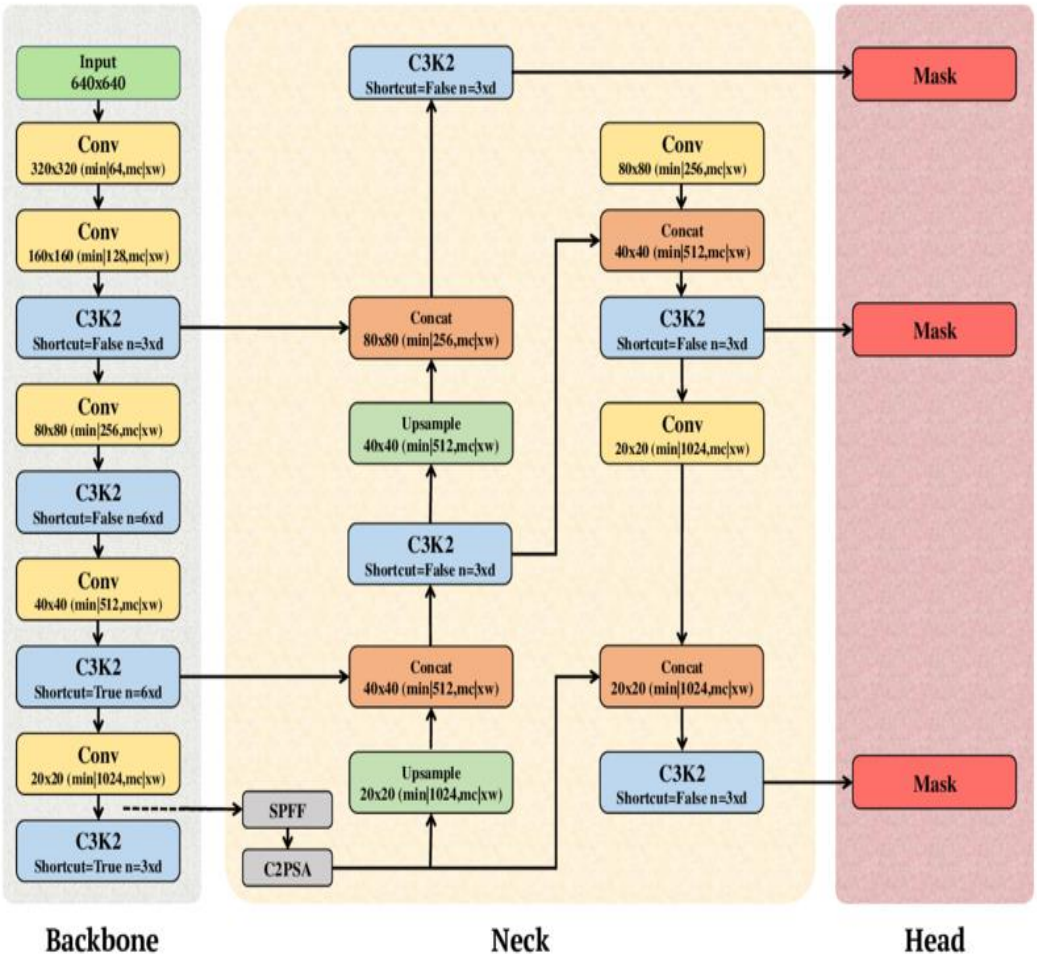

Figure 3-YOLOv11 Architecture

There are three main parts of a YOLOv11 model the backbone is used for extraction of initial features in the image using convolution layers and utilizes blocks like Spatial Pyramid Pooling and C3k2, the neck is used to refines and combine those features and to processes further extracted by backbone, and the head produces the final results like class probabilities and bounding boxes.

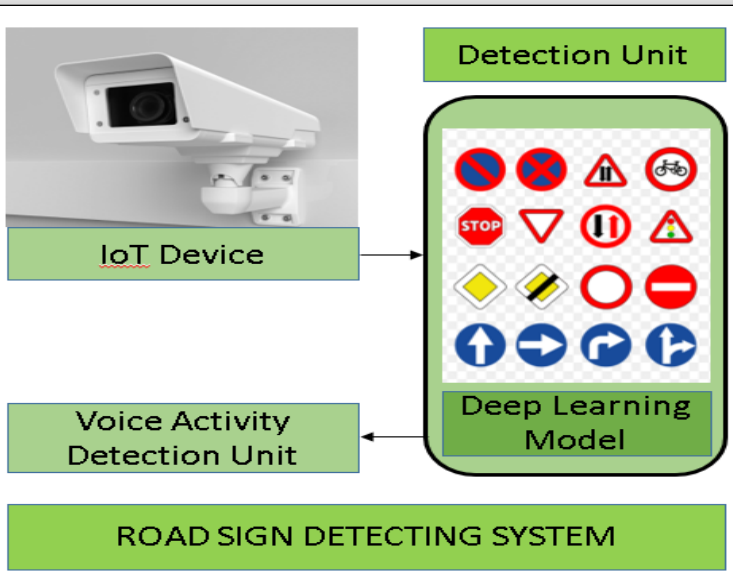

Figure 4-Road Sign Detecting System

Model Working and Construction:

The proposed model comprises of three main parts: The IoT Input unit, the Detecting unit, and the Voice activity detection. The IoT Input unit is used to capture the image; after that, it transfers the image to the detection unit. The detecting unit is used for extracting the image features, and the processed extracted feature is transferred to the Voice activity detection unit. The voice activity detection unit is used to produce the voice informing the drivers about the sign to take action accordingly.

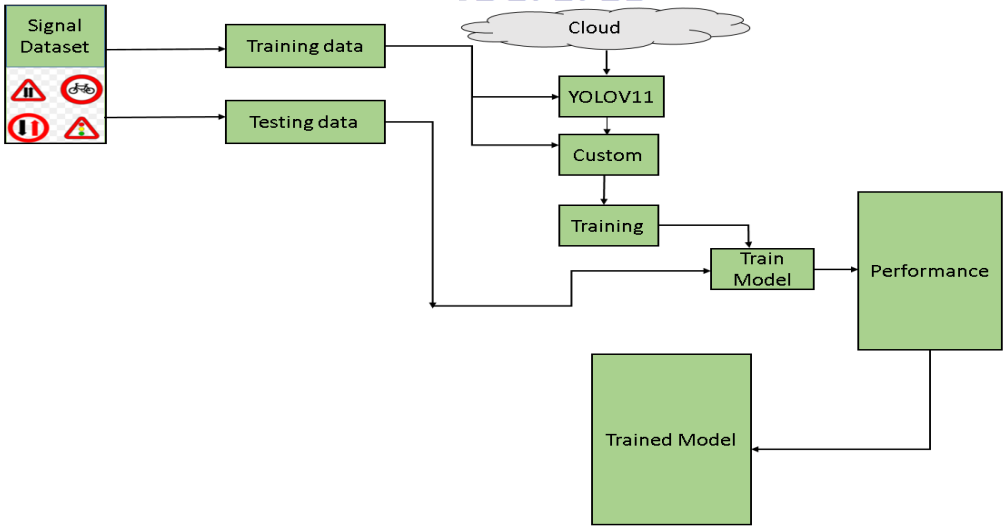

Figure 5-Trained Model

Methodology:

Dataset and Resource:

The Dataset is used in this study was downloaded from Robo-flow platform. A well-known cite for Computer vision datasets.

Data Acquisition:

The raw images were given to the proposed model for training. Different types of images are used to train model on different diverse environment to increase the model robustness and generalization capabilities in real-world scenarios. Robo-flow

provides the annotated images. Annotated images will increase the training of models.

Preprocessing and Augmentation:
To increase the range and robustness of the datasets, Preprocessing and augmentation techniques were used so that model can get the maximum feature to perform the tasks.

Preprocessing and Augmentation

To enhance the diversity and robustness of the dataset, several preprocessing and augmentation techniques were applied within Robo-flow. In preprocessing resizing, auto-orient, grayscale, auto-contract adjustment is used. In augmentation we rotate the images horizontally, vertically, randomly. Cropping, shear, noise etc also done in augmentations. This is used to prevent data leakage.

Data splitting:
After preprocessing and augmentation, the data set is splits into three parts. 70% images for training, 20% for validation and 10% for testing are used.

Simulated Training Results:

Performance Metrics: These metrics are used to calculate the performance.

Table 1: Performance Metrics

Metric	Formula	Explanation
Precision	$\frac{tp}{tp + Fp}$	Measure the correctness of the model
Recall	$\frac{tp}{tp + Fn}$	Handle the classification problems of the classes
Accuracy	$\frac{tp + tn}{tp + tn + Fp + Fn}$	Measure the correctness of a classification model.
F-1Score	$2 * \frac{P * R}{P + R}$	Measure the harmonic mean between Precision and Recall of the model.
FNR	$\frac{Fp + Fn}{tp + tn + Fp + Fn}$	Predicts how the system fails to identify the positive instances of.
TPR	$\frac{tp}{tp + Fn}$	Measured the positive cases that are recognized by the classification model.
TNR	$\frac{tn}{tn + Fp}$	Measure the negative cases that are recognized by the classification model.

Confusion Matrix: It is a table that indicates how healthy a model calculates classes compared to actual results. It is used to calculate the performance of the model by showing correct and incorrect predictions.

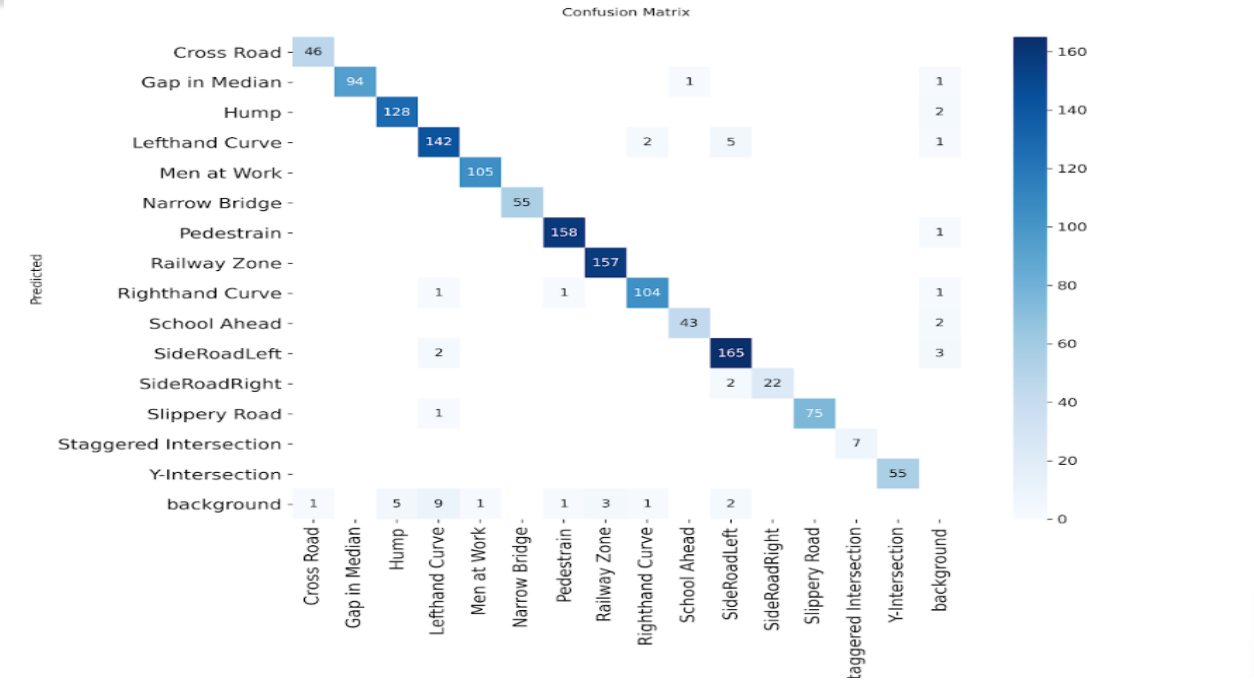

Figure 6: Confusion Metrix

Actual \ Predicted	Cross Road	Gap in Median	Hump	Lefthand Curve	Men at Work	Narrow Bridge	Pedestrian	Railway Zone	Righthand Curve	School Ahead	SideRoadLeft	SideRoadRight	Slippery Road	Staggered Intersection	Y-Intersection	Background
Cross Road	46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Gap in Median	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hump	0	0	128	0	0	0	0	0	0	0	0	0	1	0	0	5
Lefthand Curve	0	1	0	142	0	0	0	0	0	0	0	0	0	0	0	2
Men at Work	0	0	0	2	105	0	0	0	0	0	0	0	0	0	0	1
Narrow Bridge	0	0	0	5	0	55	0	0	0	0	0	0	0	0	0	1
Pedestrian	0	0	0	0	0	0	158	0	0	0	0	0	0	0	0	1
Railway Zone	0	0	0	0	0	0	0	157	0	0	0	0	0	0	0	3
Righthand Curve	0	0	0	0	0	0	0	1	104	0	0	0	0	0	0	1

Actual \ Predicted	Cross Road	Gap in Median	Hump	Left hand Curve	Men at Work	Narrow Bridge	Pedestrian	Railway Zone	Right hand Curve	School Ahead	SideRoadLeft	SideRoadRight	Slippery Road	Staggered Intersection	Y-Intersection	Background
School Ahead	0	0	0	0	0	0	0	0	1	43	0	0	0	0	0	2
SideRoadLeft	0	0	0	0	0	0	0	0	0	2	165	0	0	0	0	0
SideRoadRight	0	0	0	0	0	0	0	0	0	0	2	22	0	0	0	0
Slippery Road	0	0	0	0	0	0	0	0	0	0	0	0	75	0	0	0
Staggered Intersection	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0
Y-Intersection	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0
Background	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2: Confusion Metrix

Recall-Confidence Curve: This is a graph that tells how recall variations occur when we alter the confidence level of a model's predictions. It helps to recognize the stability between recall and confidence in classifying or detecting objects.

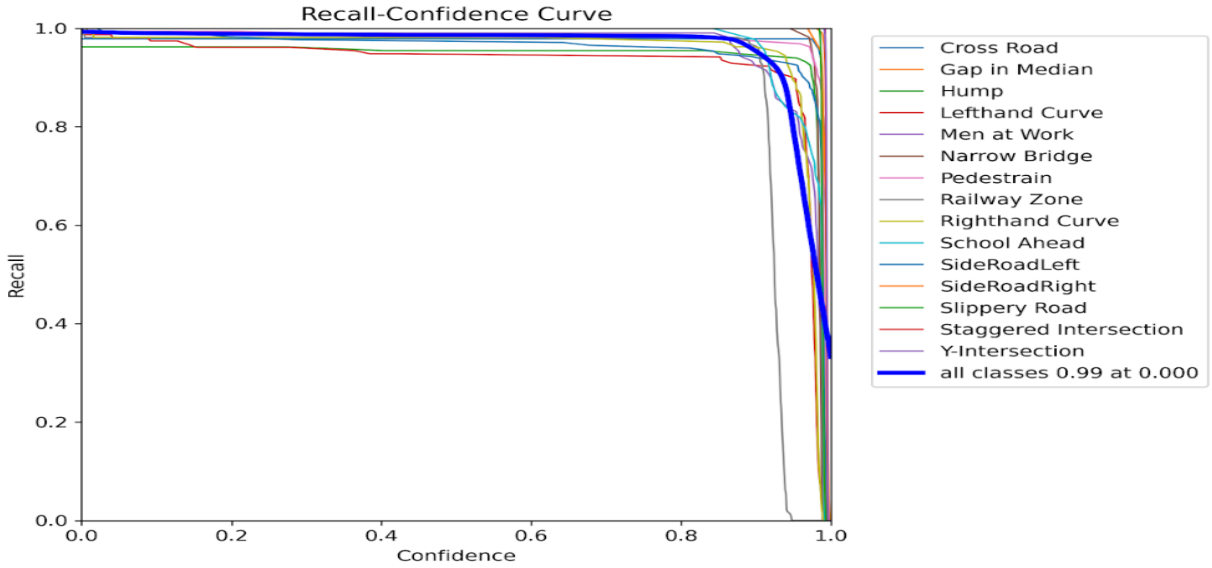

Figure 7: Recall Confidence Curve:

Precision-Recall Curve: It provides information related to **precision** and **recall** at different confidence levels. It supports calculating the performance of a model if the datasets are imbalanced.

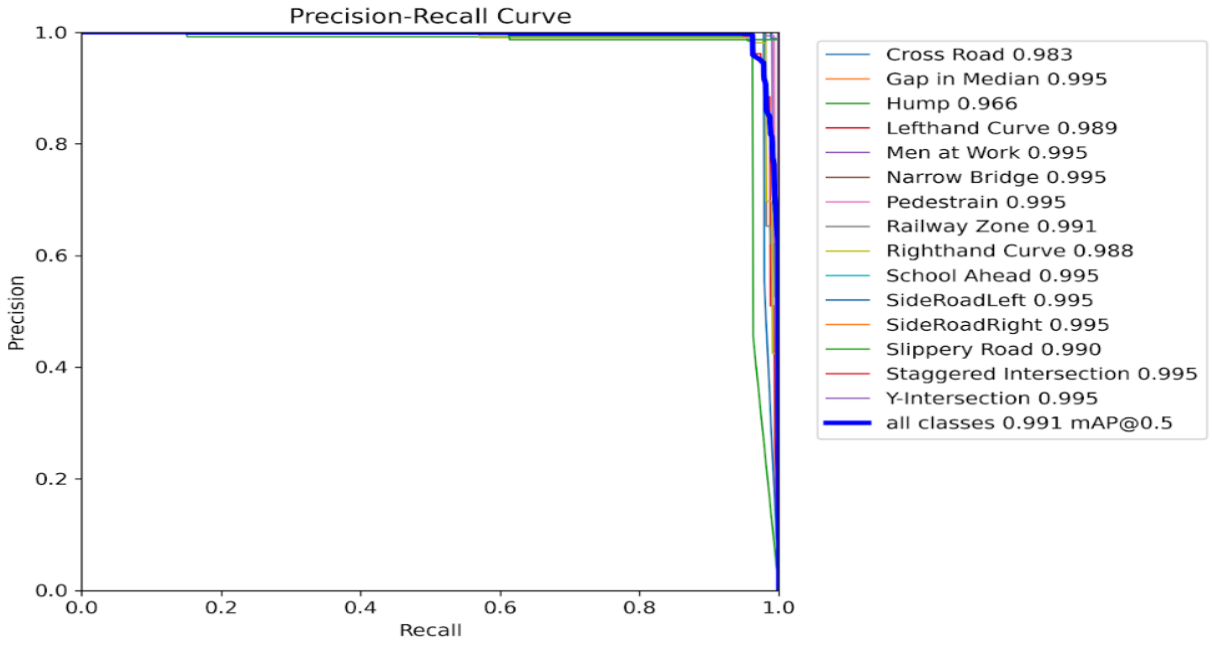

Figure 8: Precision Recall Curve

Precision-Confidence Curve: It is a graph that tells how **precision** varies as the **confidence threshold** is regulated in a model. It supports examining the adjustment between precision and confidence levels in object detection or classification.

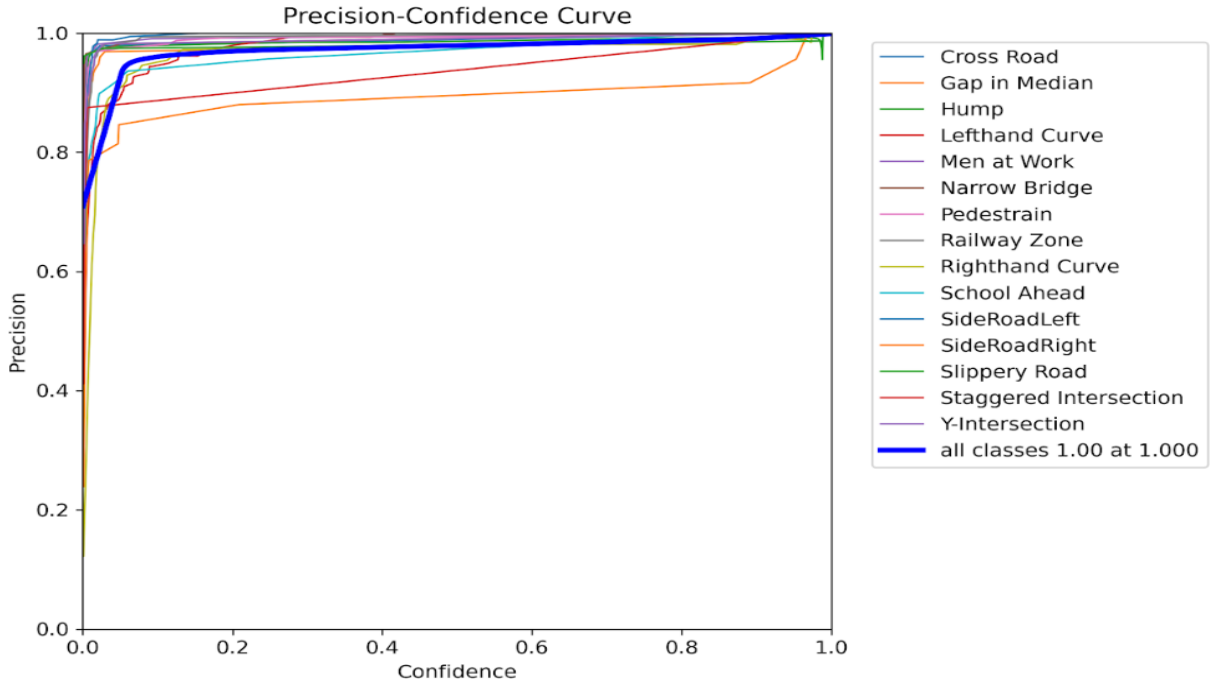

Figure 9: Precision Confidence Curve

F1-Confidence Curve: The F1-confidence curve assists you in recognizing how to balance accuracy and recall to acquire the top performance from a model.

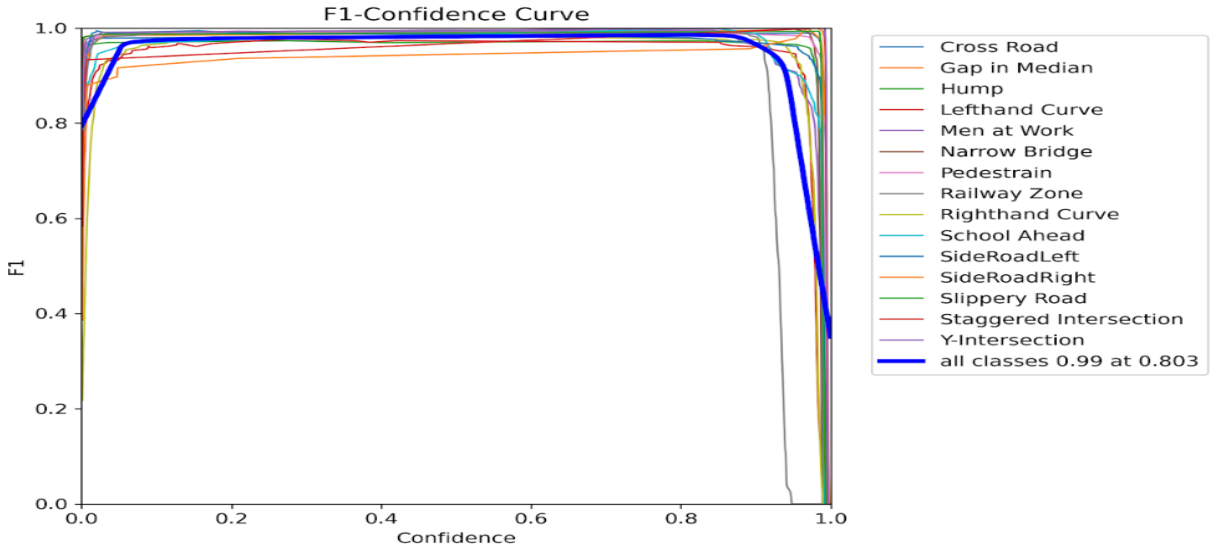

Figure 10: F-1 Confidence Curve

Results:

The results graphs represent the validation and training performance of 100 epochs.

- **Positive Learning:** The model is knowledge successfully, as exposed by decreasing losses (box, classification, and distribution focal loss) and growing precision/mAP (mean average precision) on both training and validation sets.
- **Good Generalization:** The validation metrics demonstrate that the model is generalizing healthy to unseen data, representing it is not overlearned.
- **Potential for Improvement:** While performance is good, more enhancements might be imaginable over more training epochs, hyper parameter alteration, data augmentation, or exploring different model architectures.

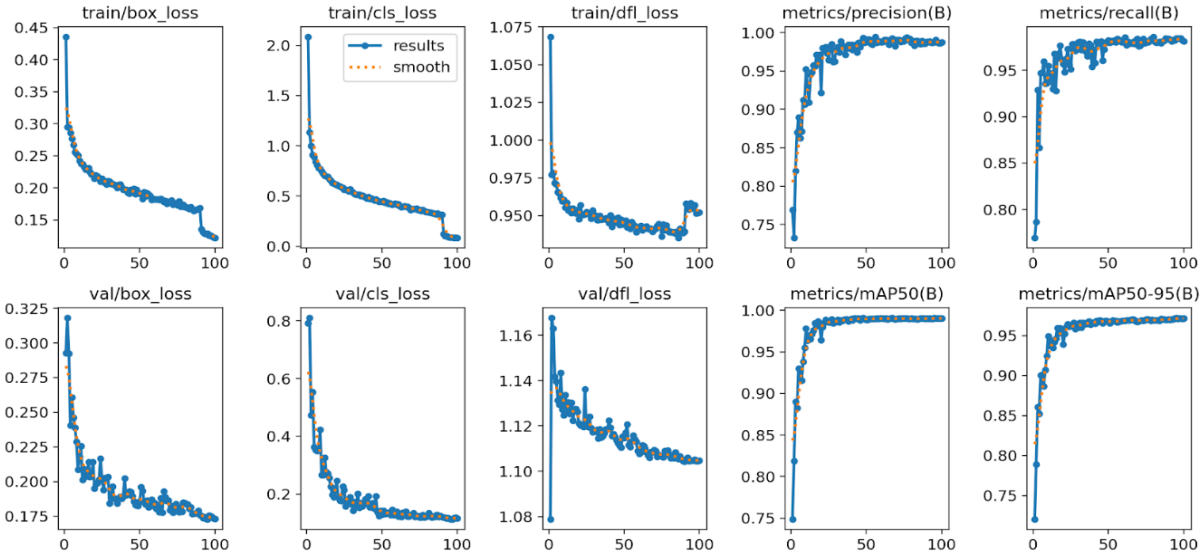

Figure 11: Results of Model during training phase

Results of different datasets on different models:

Sr.No	Model Name	Precision	Recall	F-1 Score	mAP	Validation Accuracy
1	CNN [23]				99	95
2	Yolo-BS [24]	87.9	80.5		90.1	
3	Yolo [25]			0.8987	98.8	
4	Yolov10n [26]	92.6	87.9		93.9	
5	MH Yolo based on Yolov10 [26]	85.5	88.5		94.5	
6	**Yolov11**	**95.55**	**0.9930**	**0.9759**	**98.93**	**Macro 96.17 Weighted 95.97 Overall 96.25**

Table 3: Results of Different Models

Conclusions:

Object detection is very important in real-time applications. Many approaches were used to detect the object. Identifying the boundaries of earlier approaches, such as slow image filtering and poor generalization of models trained on exact weather conditions, our approach influences the speed, accuracy, and efficiency of YOLOv11. We are proposing a Deep Learning Model (YOLOv11) to increase traffic sign detection and identification, mostly in challenging climate conditions. YOLOv11's versatility in handling numerous computer vision tasks, its flexibility to dissimilar conditions and devices, and its completely enhanced performance mark it as an encouraging tool for real-time traffic sign detection. The proposed model comprises an IoT input unit for image capture, a detection unit utilizing YOLOv11 for feature extraction and processing, and a voice activity detection unit to provide drivers with auditory alerts regarding detected signs. This integrated system aims to increase driving safety and traffic efficiency by permitting vigorous and consistent traffic sign detection in real-world driving scenarios.

REFERENCES

[1] Z.-Q. Zhao, P. Zheng, S.-T. Xu, and X. Wu, "Object Detection With Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020, doi: 10.1109/ACCESS.2020.2983149.

[3] S. Li, S. Wang, and P. Wang, "A Small Object Detection Algorithm for Traffic Signs Based on Improved YOLOv7," 2023. doi: 10.3390/s23167145.

[4] M. Y. Radzak, M. Alias, and M. Ahmad, *Study on Traffic Sign Recognition*. 2015.

[5] A. de la Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road traffic sign detection and classification," *IEEE Trans. Ind. Electron.*, vol. 44, no. 6, pp. 848–859, 1997, doi: 10.1109/41.649946.

[6] M. Q. Kheder and A. A. Mohammed, "Improved traffic sign recognition system (itsrs) for autonomous vehicle based on deep convolutional neural network," *Multimed. Tools Appl.*, vol. 83, no. 22, pp. 61821–61841, 2024.

[7] S. Venkatachalam, K. Krishnamoorthy, J. Srilla, S. Bharathi, S. Velusamy, and S. D, *Deep Learning - Based Farm Disturbance Bird Detection*. 2023. doi: 10.1109/ICSSAS57918.2023.10331906.

[8] T. P. Dang, N. T. Tran, V. H. To, and M. K. Tran Thi, "Improved YOLOv5 for real-time traffic signs recognition in bad weather conditions," *J. Supercomput.*, vol. 79, no. 10, pp. 10706–10724, 2023.

[9] H. Lai, L. Chen, W. Liu, Z. Yan, and S. Ye, "STC-YOLO: Small object detection network for traffic signs in complex environments," *Sensors*, vol. 23, no. 11, p. 5307, 2023.

[10] J. Yu, X. Ye, and Q. Tu, "Traffic sign detection and recognition in multiimages using a fusion model with YOLO and VGG network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16632–16642, 2022.

[11] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv Prepr. arXiv2410.17725*, 2024.

[12] T. Ye *et al.*, "YOLO-FIX: Improved YOLOv11 with Attention and Multi-Scale Feature Fusion for Detecting Glue Line Defects on Mobile Phone Frames," 2025. doi: 10.3390/electronics14050927.

[13] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the Performance of One-Stage and Two-Stage Object Detectors in Autonomous Vehicles Using Camera Data," 2021. doi: 10.3390/rs13010089.

[14] J. Khan *et al.*, "Can Machine Learning Enhance Intrusion Detection to Safeguard Smart City Networks from Multi-Step Cyberattacks?," 2025. doi: 10.3390/smartcities8010013.

[15] P. Bharati and A. Pramanik, "Deep Learning Techniques—R-CNN to Mask R-CNN: A Survey," 2020, pp. 657–668. doi: 10.1007/978-981-13-9042-5_56.

[16] R. B. Girshick, "Fast R-CNN, 2015. CoRR, abs/1504.08083," *URL http//arxiv. org/abs/1504.08083*.

[17] Y. Nagaoka, T. Miyazaki, Y. Sugaya, and S. Omachi, "Text Detection Using Multi-Stage Region Proposal Network Sensitive to Text Scale," 2021. doi: 10.3390/s21041232.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[20] L. Tan, T. Huangfu, L. Wu, and W. Chen, "Comparison of RetinaNet, SSD, and YOLO v3 for real-time pill identification," *BMC Med. Inform. Decis. Mak.*, vol. 21, Nov. 2021, doi: 10.1186/s12911-021-01691-8.

[21] M. L. Ali and Z. Zhang, "The YOLO Framework: A Comprehensive Review of Evolution, Applications, and Benchmarks in Object Detection," 2024. doi: 10.3390/computers13120336.

[22] Z. He, K. Wang, T. Fang, L. Su, R. Chen, and X. Fei, "Comprehensive Performance Evaluation of YOLOv11, YOLOv10, YOLOv9, YOLOv8 and YOLOv5 on Object Detection of Power Equipment," *arXiv Prepr. arXiv2411.18871*, 2024.

[23] K. Sakthivel, B. Raghul, and E. Raghul, "Traffic sign recognition system using CNN and Keras," *Int. J. Health Sci. (Qassim).*, pp. 4986–4994, Jun. 2022, doi: 10.53730/ijhs.v6nS4.9226.

[24] H. Zhang, M. Liang, and Y. Wang, "YOLO-BS: a traffic sign detection algorithm based on YOLOv8," *Sci. Rep.*, vol. 15, no. 1, p. 7558, 2025, doi: 10.1038/s41598-025-88184-0.

[25] Y. Wu, T. Zhang, J. Niu, Y. Chang, and G. Liu, "YOLO-based lightweight traffic sign detection algorithm and mobile deployment," *Optoelectron. Lett.*, vol. 21, no. 4, pp. 249–256, 2025, doi: 10.1007/s11801-025-4153-2.

[26] Y. Zhang, H. Ma, C. Zhang, Z. Wang, and Z. Li, "MH-YOLO: A Lightweight Traffic Sign Detection Method Based on YOLOv10n with Hybrid Attention Transformer and Multi-Scale Dilated Attention," 2025.