Real-Time Traffic Signs Recognition for Smart Vehicles

Ahmad Zaghameem
Computer Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
ahm20248106@std.psut.edu.jo

Amjed Al-Mousa
Computer Engineering Department
Princess Sumaya University for Technology
Amman, Jordan
a.almousa@psut.edu.jo

***Abstract*—The advancement of autonomous vehicles has increased the need for efficient and accurate real-time traffic sign recognition to ensure road safety, support driver assistance features, and enhance decision-making. This project proposes a deep learning model specializing in real-time traffic sign detection and recognition. A real-world dataset is used for training and validation. The model performed well and achieved a good F1-score.**

Keywords—Traffic sign, Recognition, YOLO, Convolutional neural network, Deep learning, Transfer learning, Ultralytics

I. INTRODUCTION

Throughout the last century, the automobile industry achieved remarkable milestones in manufacturing reliable, safe, and affordable vehicles. Because of significant recent advances in computation and communication technologies, autonomous cars are becoming a reality [1]. Cars such as the Tesla Model S and the Volvo XC90 now feature advanced self-driving functions, with tens of thousands of these vehicles on roads worldwide and more appearing every year. In addition, Tesla and other companies like Delphi and Google are testing fully autonomous cars, which have traveled millions of miles on American roads [2].

The wide deployment of autonomous vehicles requires a reliable and accurate recognition model, as Autonomous cars' ability to drive effectively and safely depends heavily on their capability to recognize traffic signs. Traditional methods are not capable of doing such a task due to varying road sign appearances, lightning changes, and complex backgrounds [3].

In recent years, a wide range of advanced object detection algorithms have been developed, such as RCNN [4], fast-RCNN [5], faster-RCNN [6], and You Only Look Once (YOLO) [7]. These methods improved traffic signs detection and recognition significantly in terms of accuracy and detection speed.

This project introduces a YOLOv8 model, a widely used Convolutional Neural Network (CNN) for real-time detection and recognition. The model was trained on a Traffic Signs dataset specifically gathered for the Africa Region from Kaggle. This project evaluates YOLOv8's performance and robustness under real-world conditions with imperfect data, while also analyzing the effects of data imbalance on the model's detection accuracy.

II. RELATED WORKS

Many researchers have applied deep learning models for traffic sign recognition. Paper [8] introduces a deep ensemble learning algorithm that uses YOLOv5s for sign detection and a MobileNet network for recognition. This method shows a strong performance, achieving an accuracy of 95.83% and a precision of 87.34%.

Paper [9] presents a comparison between Local Binary Pattern (LBP), CNN, and Transfer Learning (TL) in traffic sign detection. The experimental results show that TL is the best approach, reaching an accuracy of 98%, while CNN and LBP combined with CNN achieved 95% and 96%, respectively.

A hybrid method for traffic sign recognition has also been presented. Paper [10] describes a hybrid deep ensemble learning technique for driving assistance systems, using deep learning for feature extraction and ensemble learning for classification. They have evaluated their model using an Indian traffic sign dataset, achieving a performance on par with state-of-the-art techniques with an accuracy of 92.10%.

Furthermore, Paper [11] presents MixChannel_YOLO, a new hybrid model designed to balance accuracy and computational efficiency. This model incorporates a MixChannel attention mechanism to enhance feature extraction and employs a Slim-Neck architecture to minimize model complexity. Evaluated on the Chinese Traffic Sign Dataset (CCTSDB2021), the model achieved an mAP@0.5–0.95 of 54.7%.

Paper [12] proposes a lightweight deep learning model, MobileNetV2, a model that balances between accuracy and computational efficiency. The total number of parameters for this model sums up to 3,656,129. The model was trained using Indian traffic signs, which were collected from Kaggle. The performance of the model proved to be good as it achieved an average accuracy of 97.6%, an average precision of 97.6%, an average recall of 97.5%, and an average F1-score of 97.55%.

III. EXPERIMENTAL SETUP

This section describes the training environment by showing the hardware, software, and the python libraries used, as well as explaining the dataset that is used to train the model

A. Training Environment

The model has been trained using the Ultralytics framework. Table 1 and Table 2 show the Specifications of the environment and the libraries that were used.

Table 1. Hardware & Software Specifications

Category	Specification
CPU	i9-13900KF
GPU	RTX 4080
RAM	32 GB
Operating System	Windows 11 Pro
Python	3.10.16
CUDA Version	12.6

Table 2. The Main Libraries Used in This Project

Library	Purpose
ultralytics	handles training, validation, inference, and export.
PyTorch	powers model training and inference
OpenCV	Image processing
PyYAML	Used to read .yaml files
Matplotlib	Plotting training metrics, confusion matrix
tqdm	Progress bars during training and evaluation
numpy	General numerical processing
multiprocessing	Allows safe script execution on Windows
os, shutil, random	Used in dataset splitting
torchvision	Image Transformation
scikit-learn	Obtain Evaluation Metrics
albumentations	Data Augmentation

B. Dataset

The dataset is extracted using two open-source datasets: Mapillary and DFG. Which can be obtained through Kaggle [13]. The dataset only contains traffic sign images in the Africa region.

This dataset consists of 76 classes with a total amount of 19,346 images. The images are a mix between regulatory, warning, complementary, and information signs.

The histogram of the training data in the proposed dataset is shown in Figure 1.

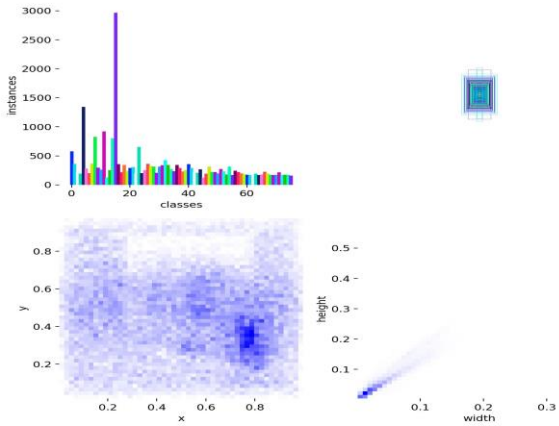

Figure 1. Training data labels distribution.

Figure 2 provides a sample preview of the dataset.

Figure 2. Samples of the dataset

The image resolution varies in range from 640x480 to 13308x6654.

The dataset has been split into train, test, and validation. The instances are shown in Table 3.

Table 3. Dataset Instances

Category	Instances
Train	15,238
Test	2,857
Val	953
Unlabeled	298

IV. Mythodology

In this section, the details of research methodology will be discussed.

A. Ultralytics Framework

The ultralytics framework made by the Ultralytics team is used to train deep learning models for different computer vision tasks. It is known for supporting YOLO models. This framework has been used in this project for its simplicity, efficiency and for providing a lot of options, including a built-in augmentation.

B. YOLOv8

The YOLO models are CNN models that are widely known for their accuracy and compact size. These models can be trained on high-end or low-end hardware [14]. YOLOv8, a state-of-the-art model that was created by the Ultralytics team, the same team that produced YOLOv5 [15], is considered a powerful model for different computer vision tasks, such as detection, classification, segmentation, and tracking. Furthermore, YOLOv8 incorporates modifications that make it powerful, namely, a revised backbone network, an anchor-free detection head, and a new loss function. Figure 3 illustrates the YOLOv8 architecture.

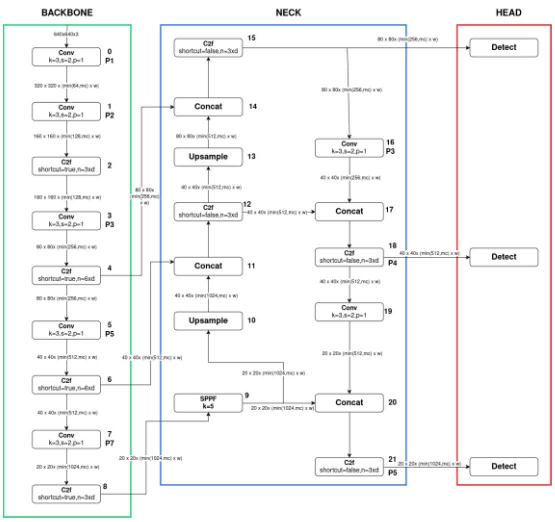

Figure 3. YOLOv8 Architecture

YOLOv8 comes in different sizes, ranging from YOLOv8.n (nano) to YOLOv8.x (xlarge), which allows a trade-off between speed and accuracy. For this project, YOLOv8.l has been used as it fite the provided data.

C. Data Augmentation

The ultralytics framework provides a built-in augmentation during the training process that applies a wide range of augmentations on images and labels by default, such as rotation, shift, and mosaic augmentation. This feature can be customized or disabled completely.

D. Training configurations

The proposed model uses AdamW [16] as its optimizer, initialized with a learning rate of 1e-2. The training was performed for 49 epochs with a batch size of 32. All images were resized to 640 x 640 before being fed into the model. To reduce overfitting, an early stopping mechanism has been activated; training will stop if no improvement is observed for 10 epochs.

E. Model Evaluation

YOLOv8 uses mAP50, mAP50-95, precision, recall, and F1-Score as its evaluation metrics. The computational methods for these metrics can be found in [17] and [18]. As for the loss function, YOLOv8 uses a composite loss function that combines multiple components: box loss, objectness loss, and classification loss. These components are added together to calculate the total loss.

V. Results and Discussion

This section presents and discusses the training results obtained from the model for detecting and recognizing traffic signs.

The proposed model reached a maximum F-score of 0.79 with a confidence threshold of 0.464, as illustrated in Figure 4

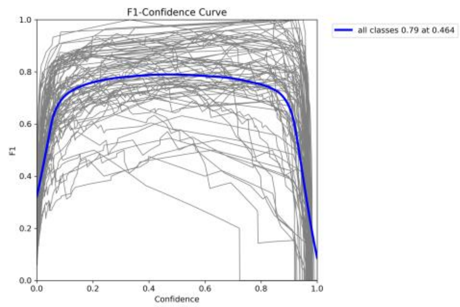

Figure 4. F1-Score Curve

The results for precision and recall are shown in Figure 5.

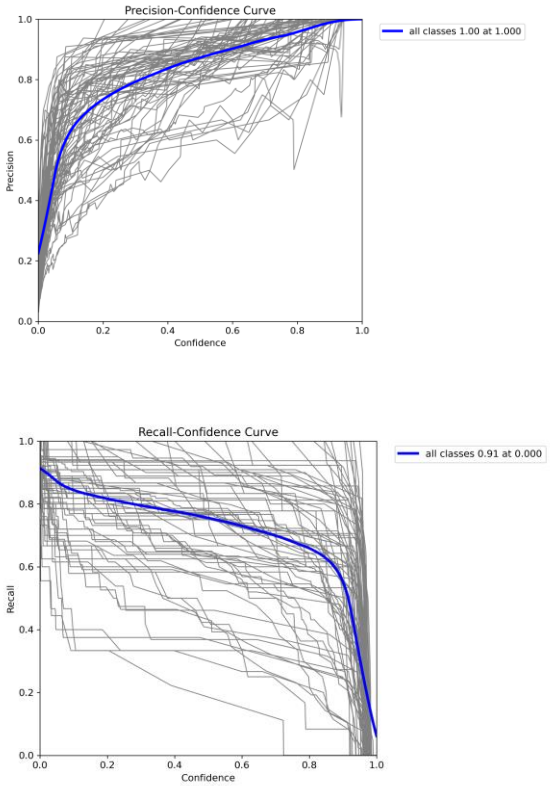

Figure 5. Precision and Recall Curves

The results for the model's performance were recorded for each epoch. Table 4 shows the results for some epochs.

Table 4. Model Results

Epochs	**Results**				
	Box Loss	***CLS_Loss***	***Precision***	***Recall***	***mAP50***
1	0.94433	2.60224	0.40332	0.23618	0.2025
5	0.69427	1.10231	0.59369	0.62093	0.62028
10	0.59537	0.79546	0.7485	0.64505	0.72522
25	0.50231	0.56046	0.83339	0.73206	0.80668
49	0.43003	0.43558	0.86295	0.76116	0.84031

Table 5 shows a comparison between the proposed model and paper [19] work

Table 5. Results Comparison

Model	**Results**			
	Precision	***Recall***	***mAP5-***	***mAP50-95***
YOLO v5 [19]	0.567	0.119	0.0944	0.0687
YOLOv7 [19] (Adam)	-	-	0.1391	0.1019
Proposed	0.862	0.761	0.8401	0.7708

Finally, the confusion matrix is shown in Figure 6.

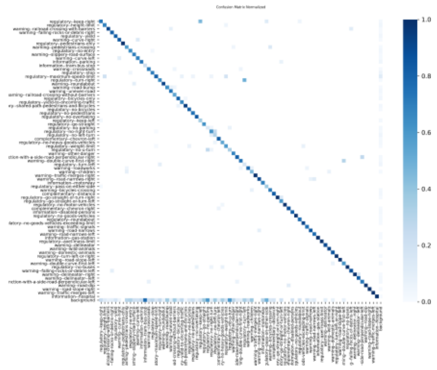

Figure 6. Confusion matrix

When taking into consideration the imbalance in the dataset and the resolution of the images. The proposed model produced good results and was able to detect and recognize traffic signs successfully for most of the classes, as shown in Figure 7.

Figure 7. Predicted Labels

Enhancing performance can be achieved through a clean and balanced dataset and parameter fine-tuning. Additionally, removing less significant classes can minimize their impact on the overall dataset.

VI. CONCLUSION

This project introduces a YOLOv8 model to detect and recognize traffic signs in real-time. Despite the dataset's imbalance and noise, the model performs good and successfully recognizes most of the classes. It achieved an average F1-score of 0.79 with a confidence threshold of 0.464. To improve the performance, some strategies can be employed, such as data cleaning by upsampling, or removing unimportant classes.

In summary, this project evaluates the performance of YOLOv8 with imbalanced and unclean real-world data.

REFERENCES

[1] R. Hussain and S. Zeadally, "Autonomous Cars: Research Results, Issues, and Future Challenges," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1275-1313, Secondquarter 2019, doi: 10.1109/COMST.2018.2869360.

[2] B. Brown, "The Social Life of Autonomous Cars," in Computer, vol. 50, no. 2, pp. 92-96, Feb. 2017, doi: 10.1109/MC.2017.59.

[3] Farhat, W., Rhaiem, O. B., Faiedh, H., & Souani, C. (2025). YOLO-TSR: A Novel YOLOv8-Based Network for Robust Traffic Sign Recognition. *Transportation Research Record*, *0*(0). https://doi.org/10.1177/03611981251327213

[4] 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). https://dl.acm.org/doi/abs/10.1109/CVPR.2014.81

[5] Girshick R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440–1448). https://ieeexplore.ieee.org/document/7410526

[6] Ren, S., He, K., Girshick, R., & Sun J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28. https://ieeexplore.ieee.org/document/7485869

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 'You Only Look Once: Unified, Real-Time Object Detection', in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[8] L. -J. Wang, T. Suwattanapunkul, J. Thalauy and P. Jansengrat, "Research on Traffic Sign Detection and Recognition System Using Deep Ensemble Learning," 2024 6th International Conference on Computer Communication and the Internet (ICCCI), Tokyo, Japan, 2024, pp. 61-66, doi: 10.1109/ICCCI62159.2024.10674294.

[9] B. Sreeja, S. Bokka, G. Shravya and K. S. Vidya Vardini, "Traffic Sign Detection using Transfer learning and a Comparison Between Different Techniques," 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-4, doi: 10.1109/ASIANCON55314.2022.9909281.

[10] A. S. Utane and S. W. Mohod, "Traffic Sign Recognition Using Hybrid Deep Ensemble Learning for Advanced Driving Assistance Systems," 2022 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Ibb, Yemen, 2022, pp. 1-5, doi: 10.1109/eSmarTA56775.2022.9935142.

[11] Y. Zhou, C. Wei, H. Liu and D. Kong, "MixChannel_YOLO: Traffic sign detection based on mix channel attention yolov8," 2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL), Zhuhai, China, 2024, pp. 1474-1478, doi: 10.1109/CVIDL62147.2024.10604111.

[12] P. Shourie, "Optimizing Traffic Sign Detection with MobileNetV2: A Lightweight Deep Learning Approach," 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, 2024, pp. 272-276, doi: 10.1109/ICTACS62700.2024.10840818.

[13] N. Ahsan, "Traffic signs dataset (mapillary and DFG)," Kaggle, https://www.kaggle.com/datasets/nomihsa965/traffic-signs-dataset-mapillary-and-dfg (accessed May 24, 2025).

[14] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, 'A review on yolov8 and its advancements', in International Conference on Data Intelligence and Cognitive Informatics, 2024, pp. 529–545.

[15] Glenn Jocher, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation". Zenodo, Nov. 22, 2022. doi: 10.5281/zenodo.7347926.

[16] Loshchilov, I. and Hutter, F., "Decoupled Weight Decay Regularization", <i>arXiv e-prints</i>, Art. no. arXiv:1711.05101, 2017. doi:10.48550/arXiv.1711.05101.

[17] H. Ling, Z. Tu, G. Li and J. Wang, "ED-YOLOv8s: An Enhanced Approach for Passion Fruit Maturity Detection Based on YOLOv8s," 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 2024, pp. 2320-2324, doi: 10.1109/AINIT61980.2024.10581681.

[18] G. R. S. N. Kumar, R. S. Sankuri and S. P. K. Karri, "Multi Scale aided Deep Learning model for High F1-score classification of fundus images based Diabetic Retinopathy and Glaucoma," 2023 International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3), Srinagar Garhwal, India, 2023, pp. 1-6, doi: 10.1109/IC2E357697.2023.10262434.

[19] A. Lok, K. Peng, and V. Wu, 'Traffic Sign Localization and Classification', 2022. [Online]. Available: https://github.com/kmpeng/mapillary-yolo.