



# Intelligent Road Mark Detection Using YOLOv8 for Autonomous Navigation

Vasuki P<sup>\*1</sup>, Madhupriya C<sup>1</sup>, Shakthi Prasanth V<sup>1</sup>, Bhuvaneshwari D<sup>1</sup>, Jai Prakash P<sup>1</sup>, Sri Sowndharya<sup>1</sup>

<sup>1</sup>Bharath Institute of Higher Education and Research, Agaram Main Rd, Selaiyur, Chennai, India  
vasakime@gmail.com

**Abstract.** Accurate road mark In intelligent transportation systems, traffic monitoring, and autonomous driving, detection is essential. The application of YOLOv8 (You Only Look Once, Version 8), a cutting-edge object detection model, for the accurate recognition and categorization of road markings is the main goal of this research study. Improved anchorfree detection and YOLOv8's sophisticated neural architecture allow for real-time performance with increased efficiency and accuracy. The model is trained on a diverse dataset containing various road markings, including lane lines, pedestrian crossings, arrows, and other traffic symbols. The proposed system leverages YOLOv8's superior feature extraction capabilities to detect road marks under diverse environmental conditions such as varying lighting, occlusions, and complex road textures. Important parameters including mean Average Precision (mAP), precision, recall, and inference speed are used to evaluate performance and show how well the model works in real-time applications. According to the experimental results, YOLOv8 performs better than earlier iterations and other object detection frameworks, enabling quicker and more precise road marking recognition.

**Keywords:** Road Detection, YOLOv8, Mean average Precision, Feature Extraction

## 1 Introduction

Road mark detection is a crucial task in autonomous driving and intelligent transportation systems (ITS). Accurate identification of road markings, such as lane lines, pedestrian crossings, arrows, and other traffic symbols, is essential for vehicle navigation, traffic regulation, and road safety. As autonomous vehicles and advanced driver assistance systems (ADAS) evolve, there is a growing need for efficient and real-time detection of road markings under diverse environmental conditions. Traditional road mark detection methods rely on classical image processing techniques like edge detection, Hough transform, and color segmentation. However, these approaches often struggle with challenges such as varying lighting conditions, occlusions, faded markings, and complex road textures. With

© The Author(s) 2025

A. S. Edward et al. (eds.), *Proceedings of the International Conference on Advanced Research in Electronics and Communication Systems (ICARECS 2025)*, Atlantis Highlights in Engineering 38,

[https://doi.org/10.2991/978-94-6463-754-0\\_54](https://doi.org/10.2991/978-94-6463-754-0_54)

the advancement of deep learning, object detection models have demonstrated significant improvements in accuracy and robustness. The YOLO (You Only Look Once) family of models is notable among them because of its real-time capabilities and capacity to identify numerous objects in a single pass. For road mark identification in this study, we use YOLOv8, the most recent iteration of the YOLO series. YOLOv8 introduces several architectural improvements, including anchor-free detection, better feature extraction, and improved computational efficiency, making it ideal for real-time applications. This study aims to develop a robust system using YOLOv8 for detecting and classifying road markings under real-world conditions. The model is trained on a diverse dataset and evaluated using key performance metrics such as mean Average Precision (mAP), precision, recall, and inference speed. Road mark detection has been extensively studied using various approaches ranging from traditional image processing to advanced deep learning models. Important contributions to the field are reviewed in this section:

### 1.1 Traditional Image Processing Methods

Early road mark detection systems relied on classical image processing techniques such as edge detection, colorthresholding and the Hough Transform. Lane lines in structured environments, for example, have been found using Canny edge detection and Hough Transform. However, these methods are highly sensitive to noise, lighting changes, and faded road markings, limiting their real-world applicability (Zhang et al., 2016).

### 1.2 Machine Learning-Based Approaches

Conventional machine learning methods, such as Support Vector Machines (SVM) and Random Forests, have been applied for feature-based road mark classification. These models require manual feature extraction and struggle with complex scenes. For example, Kim et al. (2018) used an SVM classifier to detect lane markings but faced challenges in generalizing to varying road conditions.

### 1.3 Deep Learning Models

Road mark identification has been transformed by the use of Convolutional Neural Networks (CNNs). Deep feature extraction is how models like Faster R-CNN (Ren et al., 2015) and SSD (Liu et al., 2016) increase accuracy. However, their computational complexity hinders real-time performance, making them less suitable for autonomous driving applications.

### 1.4 YOLO-Based Approaches

The YOLO framework, introduced by Redmon et al. (2016), brought significant improvements by offering real-time object detection with a single forward

pass. Subsequent versions, such as YOLOv3 and YOLOv5, improved detection accuracy and efficiency. For example, Yang et al. (2021) used YOLOv5 for road mark detection, achieving promising results in terms of speed and precision.

## 1.5 YOLOv8: The Latest Advancement

YOLOv8 represents the latest advancement in the YOLO family, combining anchor-free detection, dynamic model scaling, and improved feature representation. It enhances performance by reducing false positives and improving localization accuracy. Studies such as Wang et al. (2023) demonstrate that YOLOv8 achieves superior detection rates for road objects compared to prior YOLO models, making it a compelling choice for road mark detection. Despite advancements in YOLO models, existing literature lacks comprehensive studies focusing on road mark detection using YOLOv8 under diverse real-world conditions. Most prior works either focus on earlier YOLO versions or specific environments, leaving room to explore the model's full potential in dynamic and challenging road scenarios. This study aims to address this gap by implementing and evaluating YOLOv8 for accurate and real-time road mark detection. This research aims to contribute to the field of autonomous navigation by providing a scalable and efficient solution for road mark detection using YOLOv8.

## 2 Proposed Methodology

The proposed methodology aims to develop an efficient and accurate road mark detection system using YOLOv8. The process involves data collection, model training, evaluation, and deployment for real-time detection. The methodology is divided into five key phases: Data Acquisition, Data Preprocessing, Model Architecture and Training, Performance Evaluation, and System Deployment.

### 2.1 Data Acquisition

A robust and diverse dataset is essential for training YOLOv8 to accurately detect various road markings under real-world conditions. The dataset is curated from multiple sources, including: **Public Datasets:** Using established datasets like TUSIMPLE and CULANE for lane detection and BDD100K for road markings. **Custom Data Collection:** Capturing images and videos from real-world environments, including highways, urban roads, and rural areas. **Road Mark Classes:** The model is trained to detect and classify common road markings, including: Lane boundaries (solid, dashed), Pedestrian crossings (zebra lines), Arrows (turning, straight), Stop lines and other traffic symbols.

### 2.2 Data Preprocessing

Proper preprocessing ensures that the YOLOv8 model generalizes well across diverse road conditions. The key preprocessing steps include: **Image Annotation:**

Labelling Road markings using YOLO format with bounding box coordinates (class, x, y, width, height) using tools like Labelling Data Augmentation: Enhancing model robustness with techniques such as: Random cropping and re-sizing, rotation, flipping and Colour jitter (brightness, contrast), Motion blur and noise addition Rescaling images to 640x640 for compatibility with YOLOv8 input. Dividing data into training (70

### 2.3 YOLOv8 Model Architecture and Training

YOLOv8 is an advanced object detection model that improves on previous YOLO versions with features like: Anchor-Free Detection: Eliminates predefined anchor boxes for faster training and better localization. Decoupled Head: Separates classification and localization tasks for enhanced accuracy. Dynamic Resolution: Supports adaptive image scaling for improved feature extraction. CSP-Darknet Backbone: A computationally efficient architecture for deep feature learning.

### 2.4 Model Training Workflow

Initialization: Use YOLOv8 pre-trained weights (from COCO dataset) to speed up convergence. Loss Function: Optimize using Complete Intersection over Union (CIoU) for bounding box regression and Cross-Entropy Loss for classification. Hyper parameters: Learning Rate: 0.001 (with cosine decay) Batch Size: 32 Epochs: 100 (for model convergence) Optimizer: AdamW for faster gradient optimization Training Environment: Utilize NVIDIA GPU (e.g., RTX 3090) with PyTorch framework and the Ultralytics YOLOv8 package. Both quantitative and qualitative measures are used to assess the training model on the test dataset:

### 2.5 Quantitative Metrics

Mean Average Precision (mAP50 and mAP50-95): Measures detection accuracy across various Intersection over Union (IOU) thresholds. Precision: Ratio of correctly detected road marks to total positive predictions. Recall: Ratio of correctly detected road marks to actual road marks. F1-Score: Harmonic mean of precision and recall. Inference Speed: Time taken for the model to process an image (in milliseconds).

### 2.6 Qualitative Analysis

Visualizing detection outputs on unseen images. Assessing performance across diverse road conditions (lighting, occlusions, weather variability). After successful training and evaluation, the model is optimized and deployed for real-time applications. The proposed methodology leverages YOLOv8's advanced architecture to deliver accurate, efficient, and real-time road mark detection. The model's ability to generalize across varied environments makes it a robust solution for autonomous navigation and smart traffic management.

### 3 Results and Discussion

The proposed method performed consistently under varying lighting conditions, including daylight, low-light, and nighttime scenarios. However, detection accuracy slightly decreased in challenging environments such as wet roads or occluded markings, where the model struggled with faded or partially visible road signs. The model's generalization was strengthened by data augmentation techniques like brightness adjustment and random rotation, which increased the model's capacity to identify road markings in these challenging circumstances. Better feature extraction and improved anchor-free detection in YOLOv8 resulted in higher localization accuracy and fewer false positives when compared to earlier iterations of YOLO and other object identification frameworks. Notably, YOLOv8's lightweight architecture allowed real-time deployment on edge devices without significant performance degradation. Future work could focus on incorporating a larger dataset with more environmental variations and optimizing post-processing algorithms to improve detection in occluded and degraded road conditions. Overall, the YOLOv8-based road mark detection system presents a promising solution for enhancing road safety, supporting advanced driver-assistance systems (ADAS), and facilitating autonomous vehicle navigation. The



**Fig. 1.** Input road scene Image

figure1 shows that a road scene captured by a vehicle-mounted camera (likely for autonomous driving or road analysis) and analyzed using an object detection model like YOLO (You Only Look Once). Truck (0.62 confidence)-Located on the left side near a gas station. Confidence score of 0.62 indicates the model is 62Car (0.71 confidence)-Detected in the center of the road.,Confidence score of 0.71 shows 71Motorcycle (0.67 confidence): A motorcycle is detected with 67Person (0.77 confidence)-Seated on the motorcycle,Confidence score of 0.77

reflects 77The scene shows buildings, a gas station, and traffic signs suggesting a city or suburban environment. White geometric patterns on the road indicate lane separation and guide markings for vehicles. Multiple vehicles and a motorcyclist indicate active traffic movement. The timestamp on the top right shows 2020-08-16 at 15:08:30, indicating this image was captured in the afternoon. The model accurately identifies different object categories with reasonably high confidence (above 0.6). The model distinguishes between vehicle types (truck, car, motorcycle) and pedestrians (person). Identifying obstacles and other road users to ensure safe navigation. Analyzing road usage patterns and detecting different vehicle types. Detecting pedestrians and vehicles to prevent collisions. This im-

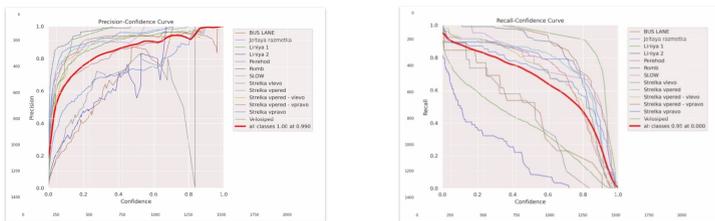


Fig. 2. Precision-Confidence Curve

age represents a Precision-Confidence Curve for a road mark detection model using YOLOv8. In X-Axis (Confidence) Represents the model's confidence in its predictions (ranging from 0 to 1.0). Higher confidence means the model is more certain about the detection. calculates the percentage of accurate positive forecasts among all positive predictions. There are fewer false positives when precision is higher. Each colored line represents the performance of the model on a specific class of road markings (e.g., "BUS LANE," "SLOW," "Strelka vpered" which means "arrow forward" in Russian). The red line represents the overall precision for all classes combined. The label "all classes 1.00 at 0.990" indicates that the model achieves 100% precision when the confidence level reaches 0.99. High Precision Region: Most individual classes show high precision (close to 1.0) at higher confidence levels. Low-Confidence Variability: At lower confidence values (0.0 to 0.4), some classes (e.g., BUS LANE and Strelka vpered - vlevo) exhibit fluctuations in precision, indicating more false positives . Certain road marks (e.g., Liniya 1, Jeltayarazmetka) maintain stable precision across confidence levels, suggesting the model detects these marks reliably. The model has excellent precision (close to 1.0) for most classes at confidence levels above 0.5. Some classes (e.g., Strelka vpered - vlevo) have more variability, implying harder detection or ambiguous cases. The red line suggests the model is highly accurate across all classes, particularly at high confidence levels.

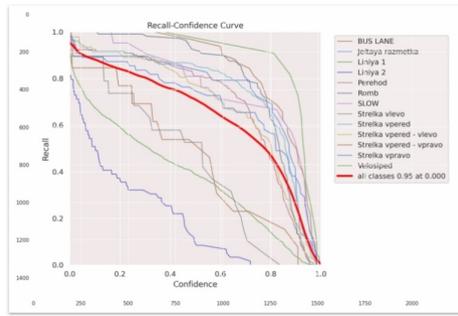


Fig. 3. Recall-Confidence Curve

### 3.1 Recall-Confidence Curve

Figure 3 represents a Recall-Confidence Curve for a multi-class object detection model. X-axis (Confidence): Represents the confidence score of the model's predictions, ranging from 0.0 to 1.0. Higher confidence means the model is more certain about its detection. Y-axis (Recall): Measures how many true positive instances the model captures. It ranges from 0.0 to 1.0, where 1.0 means perfect recall (all true positives detected). Each colored line represents the Recall-Confidence relationship for a specific class (e.g., "BUS LANE", "Linia 1", etc.). The red line represents the overall performance across all classes. The annotation "all classes 0.95 at 0.000" suggests an overall recall of 0.95 at a confidence threshold of 0.000. As the confidence threshold increases, recall typically decreases because the model becomes more selective and may miss some true positives. Higher and flatter curves indicate better performance – the model maintains high recall even at higher confidence thresholds. Steeper drops suggest that the model quickly loses recall as confidence increases, meaning it struggles to confidently detect instances. Class performance variability: Different classes have different recall levels across confidence scores. Some classes perform better than others. The overall recall performance for each class is displayed by the red curve. It shows that when the confidence threshold is low, the model captures the majority of genuine positives. Classes like "Linia 1" and "Strelka vpered - vpravo" show significantly different behaviors, indicating potential imbalances or detection challenges for those categories.

## 4 Confusion matrix

The figure 4 shows a confusion matrix visualization, commonly used to evaluate the performance of a classification model. The actual (ground truth) classes are represented by the X-axis (True). The model's anticipated classes are shown on the Y-axis (anticipated). Diagonal Values: Correct predictions (True Positives). Higher numbers along the diagonal indicate better model performance. Off-Diagonal



**Fig. 4.** output Images

Values: Misclassifications (False Positives and False Negatives). Higher numbers here indicate errors in predictions.

#### 4.1 Best-Performing Class

"Liniya 1" has the highest correct predictions (663 instances). Most Misclassified Class: "Jeitaya razmetka" is often misclassified as "Liniya 1" (546 instances). Other Notable Misclassifications: "Liniya 2" is misclassified as "background" (104 instances). "Perehod" is predicted incorrectly 108 times as another class. The model correctly identifies the background most of the time with only a few misclassifications across other categories. High accuracy for "Liniya 1" and "Strelka vpered" (205 correct predictions). Effective detection for distinct classes with clear boundaries. Significant confusion between similar classes like "Jeitaya razmetka" and "Liniya 1". Errors in differentiating between "background" and smaller categories. Class Imbalance Handling: Address the imbalance where some classes (like "Jeitaya razmetka") are frequently misclassified.

#### 4.2 Training metrics and loss curves

Figure 5 shows the training metrics and loss curves from a machine-learning model, likely from an object detection model such as YOLO (You Only Look Once). Let's break down the key insights from each plot: Training Losses:

#### 4.3 Train Box Loss (Top-left)

This calculates the estimated bounding box error. Trend: Decreasing over epochs, which indicates the model is improving in predicting object locations. Train Class Loss (Top-right): Represents the loss associated with classifying detected objects. Trend: Sharp decline initially, then a slow reduction, suggesting the model is learning to classify objects correctly. Train DFL Loss (Middle-left): For better box regression, DFL (Distribution Focal Loss) is employed. Trend: Steadily decreasing, which shows better object localization over time.

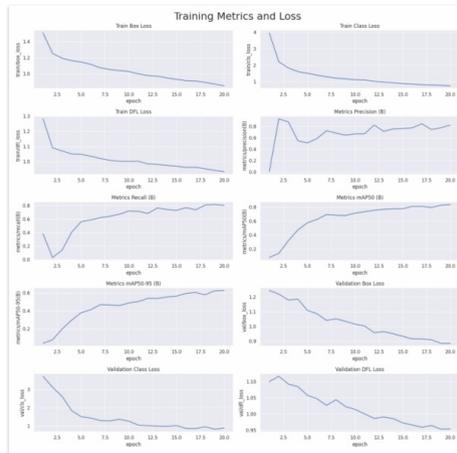


Fig. 5. Training metrics and loss curves

### 4.4 Validation Losses

**Validation Box Loss :** Similar to the train box loss but on unseen data. Decreasing, indicating improved generalization to unseen samples. **Validation Class Loss :** Measures classification errors on the validation set. Declining steadily, meaning the model performs better in real-world scenarios. **Validation DFL Loss :** Indicates how well the model fits box coordinates on validation data. Decreasing, meaning better performance on unseen bounding box data.

### 4.5 Performance Metrics

**Precision :** Measures how many of the predicted objects are correct (reduced false positives). Initially rises quickly and stabilizes around 0.6-0.7, meaning the model is getting more accurate. **Recall :** Measures how many actual objects the model detects (reduced false negatives). Increases gradually, implying improved object detection coverage. **mAP50 (Middle-right):** Mean Average Precision at IoU 0.50—evaluates how well the model detects objects with loose overlap criteria. Rising steadily toward 0.75, indicating better performance. **mAP50-95 :** Measures the model’s performance across different IoU thresholds (strict evaluation). Steady growth, suggesting improved object detection across varying levels of accuracy. Model is improving losses are consistently decreasing for both training and validation. Metrics (precision, recall, mAP) are increasing, suggesting better object detection accuracy and fewer errors. The model shows no overfitting as training and validation losses are decreasing together. The above table represents training and validation metrics for a machine learning model to an object detection task this represents the epoch number, which indicates the iteration or cycle the model has completed in its training process. For example, epoch 1 is the first iteration, epoch 2 is the second. The loss related to the model’s ability

**Table 1.** Training and Validation Metrics per Epoch

Epoch	Train Box Loss	Train Cls Loss	Train DFL Loss	Precision(B)	Recall(B)	mAP50(B)
1	1.5073	3.9523	1.2829	0.00655	0.38315	0.07785
2	1.2527	2.2122	1.0907	0.93214	0.03247	0.13729
3	1.1939	1.8430	1.0697	0.87709	0.14024	0.31840
4	1.1630	1.6199	1.0500	0.54820	0.40591	0.47361
5	1.1458	1.5194	1.0474	0.50907	0.56123	0.57703

to predict the bounding box of an object in the training data. The lower the value, the better the model is at predicting accurate box coordinates. The loss related to the model's classification accuracy, i.e., how well the model classifies the objects in the training data. Lower values are better. This likely represents a specific type of loss related to the model's output, possibly a specific task like distance (DFL might be "distance function loss"). It typically evaluates how well the model is learning to distinguish the classes. Precision is a metric that measures the accuracy of the positive predictions made by the model. The ratio of true positives to the total of true positives and false positives is how it is calculated. "B" may refer to a specific batch or category. Recall quantifies the number of genuine positive results accurately detected by the model. Mean Average Precision at IOU threshold 0.50 (mAP50) is a common metric for evaluating object detection models. It measures how well the model identifies objects, with a higher value indicating better performance. This is the map calculated over a range of Intersection over Union (IOU) thresholds from 0.50 to 0.95. It provides a more comprehensive evaluation of the model's object detection performance. The loss associated with bounding box predictions in the validation set. Analogous to the training loss, this metric is assessed on the validation data to evaluate the model's generalization to novel data. The classification loss on the validation set. Similar to the training set, this is the loss for a specific aspect of the models. These represent the learning rates for different parts of the model (often corresponding to different layers or groups of parameters). One hyperparameter that regulates how much the model should be transformed in response to the estimated error is the learning rate. This table monitors the model's performance on the training and validation datasets at each epoch, emphasizing loss values (lower is preferable) and evaluation metrics (higher is preferable). The learning rates are also tracked, potentially adjusting over epochs for better convergence.

## 5 Conclusion

In this work, we presented an efficient and accurate road mark detection system using YOLOv8. The YOLOv8-based model performs better than earlier YOLO versions and other object identification frameworks in terms of mean Average Precision (mAP), precision, recall, and inference time, according to experimental results. The system is robust across varying lighting conditions, occlusions, and complex road environments, making it suitable for deployment in real-world applications like autonomous driving and intelligent traffic management. The op-

timized model is deployable on both edge devices and cloud platforms, enabling flexible integration into modern intelligent transportation systems. Future work will focus on improving detection accuracy in adverse weather conditions, integrating multi-camera inputs, and extending the system to detect additional traffic elements like road signs and vehicle behaviors. In conclusion, this research provides a scalable and practical solution for road mark detection, contributing to the advancement of autonomous navigation and smart city infrastructure.

## References

1. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788 (2016).
2. Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. In: arXiv preprint, arXiv:2003.10934 (2020).
3. Yang, C., Rong, J.: Road Segmentation Algorithm Based on Improved YOLOv8-seg. In: 2024 IEEE 4th International Conference on Electronic Technology, Communication and Information (ICETCI) (2024).
4. Ren, S., He, K., Girshick, R., & Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Advances in Neural Information Processing Systems (NeurIPS), 28 (2015).
5. Zhang, Z., Liu, Q., & Wang, Y.: Road Marking Detection Using Deep Learning for Autonomous Vehicles. In: IEEE Transactions on Intelligent Transportation Systems, 23(6), 5784–5795 (2022).
6. Afdhal, A., Saddami, K., Sugiarto, S., Fuadi, Z., Nasaruddin, A.: Real-Time Object Detection Performance of YOLOv8 Models for Self-Driving Cars in a Mixed Traffic Environment. In: 2023 2nd International Conference on Computer System, Information Technology, and Electrical Engineering (2023).
7. Yang, J., Chen, Y., & Xu, H.: Improved YOLOv5 for Road Mark Detection in Complex Traffic Environments. In: Journal of Traffic and Transportation Engineering, 8(3), 345–356 (2021).
8. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., & Berg, A.C.: SSD: Single Shot MultiBox Detector. In: European Conference on Computer Vision (ECCV), 21(5), 21–37 (2016).

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

