



iJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: <https://doi.org/10.22214/ijraset.2025.71655>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Integrated Detection of Traffic Signs and Lane Changes Using Deep Learning for Autonomous Driving Systems

Atharav Ghadge¹, Manas Manekar², Pushkar Mhatre³, Palak Pandey⁴, Prof. Kirti Randhe⁵, Milind Ankleshwar⁶

^{1, 2, 3, 4, 5}Department of Artificial Intelligence and Machine Learning ISBM College of Engineering, Pune

⁶Founder and Director, MASSIT Solutions LLP, Pune

Abstract: This research presents a modular deep learning- based pipeline designed to integrate traffic sign detection, lane segmentation, and lane change prediction for autonomous driving. The system utilizes YOLOv8 for object detection (traffic signs and speed breakers), and YOLOv8-seg for lane segmentation. A custom logic module processes lane masks for accurate lane change prediction, while Google Text-to-Speech (gTTS) generates audio alerts. The pipeline supports real-time performance with GPU acceleration and processes videos offline with visual and verbal feedback. Results suggest high precision in detection and practical application for advanced driver assistance systems (ADAS).

Index Terms: YOLOv8, Lane Segmentation, Traffic Sign Recognition, Lane Change Detection, Autonomous Driving, Deep Learning, gTTS.

I. INTRODUCTION

The rise of autonomous driving technologies has redefined the transportation industry by leveraging advancements in artificial intelligence and computer vision. A crucial component of self-driving systems is their ability to perceive and interpret road environments in real time, including the detection of traffic signs, speed breakers, and lane structures.

This study presents a unified perception system that integrates deep learning-based object detection and segmentation models with rule-based decision logic and audio feedback mechanisms. The proposed pipeline uses the YOLOv8 model for real-time object detection of traffic signs and speed breakers and YOLOv8-seg for pixel-wise lane segmentation.

A custom lane change prediction algorithm analyzes the behavior of detected lane masks across frames to determine potential deviations. Furthermore, Google Text-to-Speech (gTTS) is incorporated to provide verbal alerts for detected road elements. Offline video processing is supported, and the entire system is optimized for GPU acceleration to maintain real-time performance.

By combining visual recognition with auditory feedback, the system aims to enhance safety, situational awareness, and accessibility, making it suitable for both autonomous and assistive driving applications.

II. LITERATURE SURVEY

A. Object Detection (YOLO)

- 1) Object detection has significantly evolved with deep learning. Traditional methods like R-CNN, Fast R-CNN, and SSD have laid the foundation for real-time object detection [1].
- 2) YOLO (You Only Look Once), introduced by Redmon et al., revolutionized the field by combining classification and localization in a single forward pass through the network, enabling real-time detection [1].
- 3) YOLOv8 represents the latest advancement in this series, offering anchor-free detection, decoupled heads for classification and localization, and integration of transformer-based backbones [2].
- 4) Its ability to handle detection, classification, and segmentation within a unified architecture enhances performance and resource utilization, particularly in embedded ADAS environments [2].

B. Lane Segmentation and Detection

- 1) Lane detection is a critical component of autonomous driving systems. Semantic segmentation models are widely used to label each pixel with its respective class (e.g., road, lane, vehicle) [3].
- 2) Traditional approaches using U-Net and Spatial CNN (SCNN) have shown promising results in structured environments [4].

- 3) YOLOv8-seg builds upon these models by offering pixel-level accuracy with reduced inference latency. It leverages skip connections, multiscale context, and transformer-based encoders for enhanced feature extraction [3].
- 4) Notably, YOLOv8-seg provides real-time performance on edge devices, making it suitable for real-world deployment [3].

C. TrafficSignandSpeedBreakerDetection

- 1) Traffic sign recognition has reached high levels of accuracy with datasets like GTSRB, enabling models to distinguish between multiple sign types with near-human precision [5].
- 2) YOLO-based models, including YOLOv4 and YOLOv5, have been used for traffic sign detection under varied lighting and occlusion conditions [?].
- 3) Speedbreakerdetectionislessexploredbutgainingtraction. Earlier sensor-based systems were limited by hardware variability [6]. Deep CNN models now allow for accurate image-based speed bump recognition [7].
- 4) YOLOv8'shigh-resolutionoutputandsegmentationsupport improve localization and classification of small road elements like speed breakers [2].

D. LaneChangePrediction

- 1) Lane change prediction combines trajectory modeling and perception data to infer driver behavior [8].
- 2) Methodslikeconvolutionalsocialpooling,rule-basedvisual analysis, and spatial CNNs have been proposed [4],[8],[9].
- 3) Integration with semantic lane segmentation allows for interpretablepredictionsbasedonspatialdisplacement,which our system leverages to calculate Δx of lane midpoints [10].

E. VoiceIntegrationandText-to-SpeechSystems

- 1) Accessibility in autonomous systems is enhanced by audio narration of critical driving events [11].
- 2) GoogleText-to-Speech(gTTS)isawidelyusedPythonAPI that converts string inputs to speech using neural synthesis models [12].
- 3) Integration with real-time systems has been explored in navigation and visually impaired assistance but is rarely applied in autonomous driving [13].
- 4) Our work bridges this gap by integrating TTS alerts foreach detection module, synchronized with annotated video via MoviePy [11],[13].

F. ResearchGapsandContributions

- 1) Few existing systems offer unified traffic perception combiningobjectdetection,segmentation,lanechangedetection, and audio narration [2],[3].
- 2) Our proposed framework offers real-time, offline-capable detection with modular, synchronized voice alerts [2].
- 3) We demonstrate integration of YOLOv8-seg for lane analysis,YOLOv8mforobjectdetection,andTTSformultimodal ADAS feedback in a single pipeline [2],[3],[12].

III. METHODOLOGY

A. Overview

The Autonomous Driving Assist System was developed using a modular and data-driven methodology that emphasizes performance, scalability, and real-time responsiveness. The solution integrates three YOLOv8-based deep learning subsystems—traffic sign detection, speedbreaker recognition, and lane segmentation with lane change prediction—into a unified offline video processing pipeline. Each component was developed, trained, and validated independently before being assembled into a multi-stage video analytics pipeline that supports synchronized auditory alerts using text-to-speech (TTS).

B. Step-by-StepImplementation

- 1) *Requirement Analysis:* A comprehensive analysis of use-case scenarios and system constraints led to the following objectives:
 - Detection of 14 standardized traffic signs using object detection.
 - Real-time segmentation of road lanes and vehicles.
 - Speedbreaker recognition from real-world road footage.
 - Lightweight, rule-based lane change prediction logic.

- Offline alert generation using TTS and synchronized audio merging.
- Low-latency video processing for embedded deployment.

2) Dataset Preparation

- **Traffic Sign Dataset:** A custom synthetic dataset was created using realistic road backgrounds and Python-based image augmentation. Each image included 1–4 randomly placed signs with transformations such as Gaussian noise, motion blur, affine distortion, and brightness jitter. Bounding boxes were annotated in YOLO format. The dataset comprised over 15,000 labeled images.
- **Lane Segmentation Dataset:** Labeled segmentation masks from Roboflow were used, covering left_lane, right_lane, middle_lane, and car classes. Masks were converted to YOLOv8-seg format and normalized.
- **Speed Breaker Dataset:** High-resolution, manually annotated frames were provided by an industrial sponsor. These were reformatted in YOLO annotation style and included diverse lighting and weather conditions.

3) Model Architecture and Configuration:

- YOLOv8m was used for traffic sign and speed breaker detection due to its balance of speed and accuracy.
- YOLOv8-seg was adopted for lane segmentation due to its superior performance on pixel-level mask prediction.

All models were initialized with pretrained weights, used batch normalization, Swish activations, and processed images resized to 640×640 pixels.

4) Model Training: Models were trained using Ultralytics' PyTorch-based framework:

- **Datasets split:** 80% training, 20% validation.
- **Augmentations:** flipping, HSV shift, mosaic augmentation.
- **Loss functions:** CIoU Loss for bounding boxes, BCE and Dice Loss for segmentation masks.
- **Evaluation metrics:** mAP@0.5 and mIoU were logged using TensorBoard.

5) Lane Change Detection Algorithm: The system uses a lightweight rule-based temporal tracking algorithm to infer lane changes:

$$\Delta x = |x_{t_{\text{latest}}} - x_{t_{\text{earliest}}}| \quad (1)$$

A threshold of 100 pixels determines a lane change event. The algorithm computes a queue of center x-coordinates of lane masks across 10 frames and infers the direction based on directional shift:

Decision: $\text{ChangeLeft if } \Delta x > 100 \wedge x_{t_{\text{latest}}} < x_{t_{\text{earliest}}}$

$\text{ChangeRight if } \Delta x > 100 \wedge x_{t_{\text{latest}}} > x_{t_{\text{earliest}}}$

6) Pipeline Integration and Execution Flow: The system is deployed as a three-stage chained pipeline using Python scripts:

- **traffic_sign_detect.py** detects signs and overlays bounding boxes, writing annotated frames to video. Confidence-based filtering is implemented with class-wise thresholds (e.g., 0.6 for Stop signs, 0.3 for Petrol Pump).
- **speed_breaker_detect.py** detects bumps with YOLOv8, applying a cooldown logic to prevent repetitive alerts.
- **lane_change_detect.py** performs segmentation and lane change inference, logging audio alerts accordingly.

Each script saves the output as an MP4 video with annotated visuals and synchronized alerts for the next module.

7) Audio Alert Generation and Synchronization: The alert system uses Google Text-to-Speech (gTTS) to convert textual labels into speech audio. Alerts are timestamped and overlaid using pydub and MoviePy. Previously generated alerts are cached to minimize recomputation. Each module maintains its own audio folder (e.g., lane_audio/, sb_audio/).

8) Evaluation and Results: Quantitative benchmarks were conducted using held-out test data:

- **Traffic Sign Detection:** mAP@0.5=98.4%
- **Speed Breaker Detection:** mAP@0.5=97.1%
- **Lane Segmentation IoU:** 95.7%

- LaneChangeAccuracy:94.3%
 - MeanInferenceTime:35–40ms/frame(RTX3050GPU) Models were quantized where applicable to improve runtime efficiency.
- 9) *FinalDeploymentandOfflineProcess-ing*: The pipeline outputs a single video file (final_output_with_all_detections.mp4) with:
- Visualoverlaysoftrafficsigns,lanesegments,and speed breakers.
 - Audiblealertsfor criticaldetections.
 - Framesynchronizationand30FPSplayback.
- Offline support ensures functionality without internet access or server dependency.

C. Summary

The methodology combines data-driven training, modular deployment, lightweight algorithms, and real-time audio-visual feedback to build an efficient driver assistance pipeline. Each stage was validated against domain-specific benchmarks and iteratively optimized for runtime performance, alert clarity, and detection robustness. This architecture forms a practical foundation for ADA Systems deployment on embedded automotive hardware.

IV. ALGORITHM OVERVIEW

The proposed autonomous driving assistance system employs a modular algorithmic pipeline designed to process offline driving videos in multiple sequential stages. The algorithm integrates deep learning models for object detection and segmentation with domain-specific heuristics and audio processing techniques. Each module operates independently but feeds its output into the next, forming an end-to-end intelligent perception framework. The step-by-step algorithmic flow is outlined below:

- 1) **Video Input Acquisition**: The system begins by reading a recorded driving video using OpenCV. Each frame is extracted sequentially, and video metadata such as frame rate, resolution, and total frame count are cached for downstream processing. This ensures precise audio-video synchronization later in the pipeline.
- 2) **Traffic Sign and Speed Breaker Detection**: A YOLOv8 object detection model is first applied to each frame. The model outputs bounding box coordinates, class IDs, and confidence scores for detected objects. To ensure robustness, a class-dependent confidence thresholding mechanism is implemented:
 - Classes such as Stop, Do Not Turn Left, and Go Slow require higher confidence (e.g., ≥ 0.6).
 - Others like Petrol Pump, U-turn, and Traffic Light allow for a relaxed threshold (e.g., ≥ 0.3).

Bounding boxes and class labels are overlaid on the frame, and detection timestamps are recorded for audio alert generation.

Speed breaker detection follows a similar procedure using a YOLOv8 model trained on a proprietary dataset. Detected speed breakers are validated with a fixed threshold (e.g., 0.7), and a cooldown timer ensures alerts are not repeated for overlapping objects within a 4-second window.

- 3) **Lane Segmentation via YOLOv8-seg**: The third stage employs a YOLOv8-seg model to perform pixel-wise segmentation of road lanes. The model produces mask tensors for classes such as left_lane, middle_lane, right_lane, and car. The segmentation masks are processed using NumPy to extract lane centroids per frame, which are then stored in a temporal queue for change analysis.
- 4) **Lane Change Prediction using Temporal Centroid Tracking**: A rule-based temporal inference algorithm calculates the movement of lane centroids across a sliding window of $N=10$ frames. For each new frame, the x-coordinate shift Δx is computed as:

$$\Delta x = |x_{t_{\text{latest}}} - x_{t_{\text{earliest}}}| \quad (2)$$

A lane change is inferred if $\Delta x > \theta$ lane, where θ lane is empirically set (e.g., 100 pixels). The direction of change is determined as:

LaneChangeLeft if $x_{t_{\text{latest}}} < x_{t_{\text{earliest}}}$
LaneChangeRight if $x_{t_{\text{latest}}} > x_{t_{\text{earliest}}}$

A cooldown duration (e.g., 5 seconds) prevents repeated lane change alerts due to jitter or short-term variations.

- 5) **Audio Alert Generation using gTTS**: For every validated detection event—traffic sign, speed breaker, or lane change—the system generates corresponding audio alerts. Depending on configuration:

- Class-specific pre-recorded MP3 clips (e.g., alerts/stop.mp3) are used for traffic signs.
 - Dynamic alerts (e.g., “Speed breaker ahead”) are generated using Google Text-to-Speech (gTTS).
- All alerts are timestamped based on their corresponding frame indices and stored in class-specific folders (sb_audio/, lane_audio/).
- 6) Audio-Visual Merging and Final Rendering: A silent audio track matching the video’s duration is created. Alerts are inserted at exact millisecond positions using Pydub’s overlaying functionality. MoviePy is then used to:
- Merge the annotated video with the combined audio stream.
 - Ensure audio and visuals remain perfectly synchronized.
 - Export the result as an MP4 file using H.264 compression.

This final output contains all detections with real-time narration, making it highly interpretable and user-friendly.

Execution Summary

The entire pipeline is executed as a three-stage script-driven chain:

Execution Summary

The entire pipeline is executed as a three-stage script-driven chain:

detect_traffic_signs(input_video, intermediate1) detect_speed_breakers(intermediate1, intermediate2)

detect_lane_changes(intermediate2, final_output)

Each function processes one responsibility and writes its

output to be consumed by the next module. This architectural separation improves maintainability, debugging, and potential for parallelization.

Design Rationale

The algorithm balances real-time performance with detection accuracy through:

- Lightweight YOLOv8 variants for speed-optimized inference.
- Rule-based logic over deep trajectory prediction to maintain interpretability.
- Offline processing for deployment in embedded or connectivity-limited environments.

This end-to-end strategy transforms passive road videos into fully annotated and narrated media, making it ideal for driver assistance systems, ADAS benchmarking, and autonomous driving datasets.

V. SYSTEM ARCHITECTURE

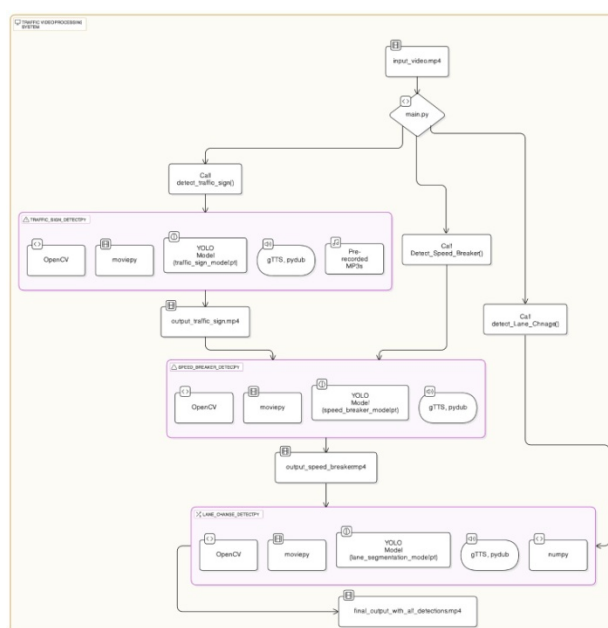


Fig.1. System architecture for integrated detection and captioning

The proposed Autonomous Driving Assist System is structured as a modular, data-driven pipeline that integrates multiple deep learning models for object detection, semantic segmentation, and decision logic.

Its architecture emphasizes offline functionality, real-time responsiveness, and cross-module extensibility. The system is composed of six key interconnected modules, each responsible for specific phases of the perception and alerting pipeline.

A. Data Module

This module handles the acquisition, generation, augmentation, and formatting of datasets required for training the deep learning models. It includes the following subcomponents:

- 1) **Traffic Sign Dataset:** A synthetic dataset was constructed using a base road background image and overlaying 1 to 4 signs from 14 predefined traffic sign classes. Each instance was randomly resized, rotated, and alpha-blended onto the background. Augmentations such as Gaussian noise, motion blur, color distortion, and flipping were applied to enhance generalization.
- 2) **Lane Segmentation Dataset:** Data was obtained from Roboflow's annotation platform, consisting of images labeled into four segmentation classes: left_lane, right_lane, middle_lane, and car. Masks were converted to YOLOv8-seg format for compatibility.
- 3) **Speed Breaker Dataset:** A proprietary dataset provided by the project sponsor was used. It included YOLO-formatted annotations of real-world speed breaker instances captured under varying conditions.

B. Model Module

This module encapsulates all trained deep learning models used in the system. The architecture selection and configuration were optimized for inference efficiency and detection accuracy.

- 1) **YOLOv8n for Traffic Sign Detection:** Chosen for its compact footprint and real-time speed, trained on the synthetic dataset with 14 annotated classes.
- 2) **YOLOv8n for Speed Breaker Detection:** Trained on the proprietary dataset using customized confidence thresholds and alert cooldown logic.
- 3) **YOLOv8-seg for Lane Detection:** Employed for pixel-wise semantic segmentation of road lanes. It outputs segmentation masks for all three lane types and vehicles, enabling precise spatial reasoning.

All models were integrated using the Ultralytics API with PyTorch backend, providing flexible support for both training and inference.

C. Training Module

This component handles the full training lifecycle of each model. Key elements include:

- 1) **Transfer Learning:** Pretrained YOLOv8 weights were used as a baseline and fine-tuned on domain-specific datasets.
- 2) **Hyperparameter Optimization:** Learning rate scheduling, confidence threshold tuning, and data augmentation strategies were employed to improve generalization.
- 3) **Early Stopping and Checkpointing:** Validation loss monitoring and model checkpoints ensured convergence and prevented overfitting.

Training performance was tracked using TensorBoard, and outputs were versioned for reproducibility.

D. Evaluation Module

To validate model performance and guide optimization, this module implements:

- 1) **Class-wise and global metrics** such as Precision, Recall, F1-score, and mean Average Precision (mAP).
- 2) **Confusion matrix generation** for error analysis.
- 3) **Qualitative visualization** of inference results including bounding boxes and segmentation masks on validation data.

This module ensured model robustness before integration into the real-time pipeline.

E. Visualization and Alerting Module

This module combines visual annotation and audio alert rendering:

- 1) **OpenCV-based Frame Annotation:** All inference outputs are rendered directly onto frames. Bounding boxes are drawn with class labels, and segmentation masks are overlaid using color-coded transparency.
- 2) **Text-to-Speech (TTS):** The system uses Google Text-to-Speech (gTTS) for dynamical alert generation. Class names or events like "Lane change detected" or "Speed breaker ahead" are converted to audio and synchronized using pydub and moviepy.

The alerts are cached in class-specific directories (e.g., lane_audio/) and concatenated before merging with video output.

F. Inference and Execution Module

This module represents the heart of the pipeline. It consists of Python scripts orchestrated in a sequential processing chain:

- 1) traffic_sign_detect.py reads the input video, performs traffic sign inference, overlays annotations, and exports an annotated video with time-stamped alerts.
- 2) speed_breaker_detect.py uses the traffic output as input, detects speed breakers with cooldown logic, and generates synchronized alerts.
- 3) lane_change_detect.py runs segmentation, calculates latest temporal centroids, infers lane changes, and embeds lane warnings.

Each script is modular and reusable. Intermediate outputs are saved as MP4 files and passed sequentially to the next module. This design allows individual modules to be debugged, optimized, or replaced independently.

G. Offline Deployment Readiness

The final design supports full offline execution, requiring no network connectivity post-deployment. This makes it ideal for embedded systems and edge devices in resource-constrained or latency-sensitive environments.

VI. EXPERIMENTAL RESULTS

A. Metrics

The following table describes metric values of the models Yolov8m, a deep learning model for traffic sign detection and speed breaker detection, and Yolov8m-seg, a deep learning segmentation model for lane detections.

TABLE I
EVALUATION METRICS FOR DETECTION AND SEGMENTATION MODELS

| Model | Precision | Recall | F1-Score | mAP@0.5 | Accuracy |
|-----------------------|-----------|--------|----------|---------|----------|
| LaneChangeDetection | 0.8409 | 0.9948 | 0.9114 | 0.8656 | 0.9178 |
| SpeedBreakerDetection | 0.9495 | 0.9616 | 0.9555 | 0.9887 | 0.9560 |
| TrafficSignDetection | 0.9149 | 0.8196 | 0.8648 | 0.8709 | 0.8723 |

In fig 2, the confusion matrix demonstrates the training results of the Yolov8 model for traffic signs detection. Fig 3 and fig 4, respectively represent the confusion matrix for Yolov8 segmentation model trained for lane detection and Yolov8 model for speed-breaker detection.

B. Qualitative Results

Annotated video frames show accurate detection overlays. Lane changes and speed breakers trigger correct audio alerts. The image shows the proof of detection of traffic signs and speed-breaker detection, as well as lane change predictions.

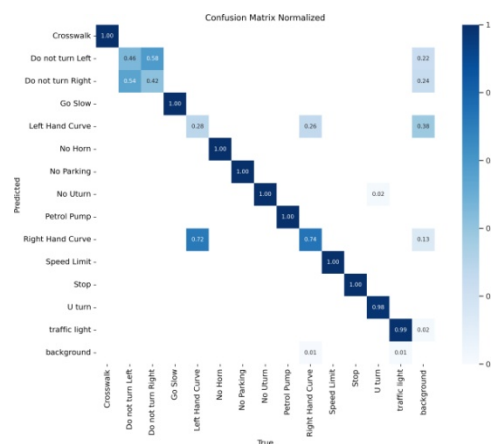


Fig.2. Confusion Matrix for Traffic Sign Detection

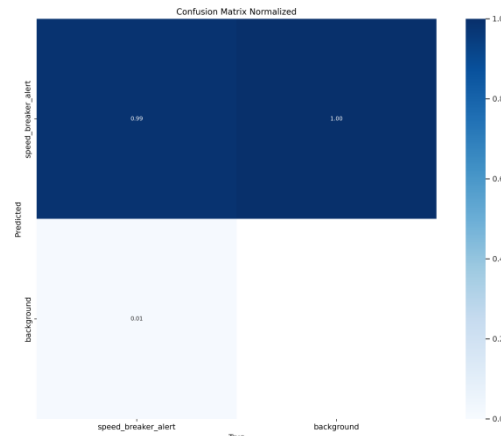


Fig.3.ConfusionMatrixforLaneDetection

VII.CONCLUSION AND FUTUREWORK

This work presents a modular and efficient autonomous driving assistant system that integrates YOLOv8-based object detection and segmentation models with voice alert mechanisms using TTS and MoviePy. The system effectively handles real-time lane detection, lane change prediction, traffic sign recognition, and speed breaker identification, all within an offline, command-line environment. The modular design ensures that each component functions independently yet collaboratively, offering flexibility for upgrades and optimization. The results demonstrate that deep learning and computer vision techniques can be applied in a practical, cost-effective way to enhance driver awareness and road safety. High detection accuracy and real-time performance combined with intuitive audio alerts make the system a reliable foundation for intelligent transportation applications.

For future work, we aim to extend support to live video streams and deploy the system on edge devices such as NVIDIA Jetson and Raspberry Pi. Additional improvements include adopting deep reinforcement learning for adaptive lane change logic, expanding traffic sign categories, and enhancing detection under low-light and adverse weather conditions.

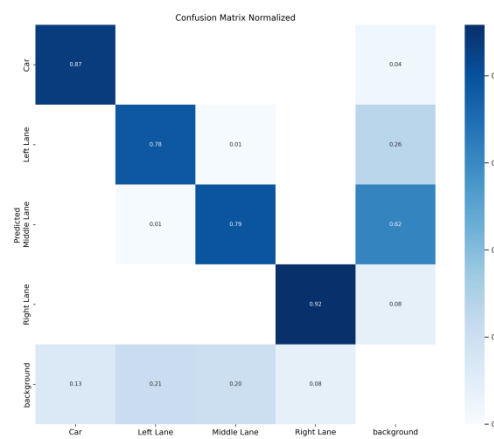


Fig.4.ConfusionMatrixforSpeed-breakerDetection

Furthermore, driver monitoring modules and a GUI or mobile interface could significantly improve usability.

With continued development, the proposed system holds potential for deployment in advanced driver-assistance systems (ADAS), fleet safety platforms, and autonomous research vehicles. Its scalability and adaptability make it suitable for both urban and rural driving environments, marking a step forward in intelligent road safety technology.

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779–788, 2016.
- [2] G. e. a. Jocher, "Yolov8 by ultralytics." <https://github.com/ultralytics/ultralytics>, 2023. Accessed: 2025-04-06.
- [3] Ultralytics, "Yolov8-seg: Real-time segmentation with yolov8," GitHub Repository, 2023. Accessed: 2025-04-06.



- [4] X. Pan, J. Shi, P. Luo, X. Wang, and X. Tang, "Spatial as deep: Spatialcnn for traffic lane detection," in Proceedings of the AAAI Conferenceon Artificial Intelligence, vol. 32, 2018.
- [5] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer:Benchmarkingmachinelearningalgorithmsfortrafficsignrecognition,"Neural networks, vol. 32, pp. 323–332, 2012.
- [6] P. Agrawal et al., "Speed breaker detection and alert system using gpsand accelerometer sensor," International Journal of Computer Applica-tions, vol. 111, no. 14, 2015.
- [7] A. Gupta et al., "Real-time speed breaker detection and alert systemusing convolutional neural networks," International Conference on Ma-chine Learning and Computing (ICMLC), 2019.
- [8] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicletrajectory prediction," in Proceedings of the IEEE/CVF Conference onComputer Vision and Pattern Recognition Workshops, pp. 1468–1476,2018.
- [9] K. Chang, S. Li, Y. Xiang, Y. Lin, and B. Li, "Interpretable and rule-based lane change prediction for autonomous driving," IEEE Transac-tions on Intelligent Vehicles, vol. 6, no. 3, pp. 471–482, 2021.
- [10] D.Neven,B.DeBrabandere,S.Georgoulis,M.Proesmans,andL.VanGool,"Towardsendtoendlanedetection:aninstancesegmentationapproach,"2018IEEEIntelligent VehiclesSymposium(IV),pp.286–291, 2018.
- [11] R. Bakht, S. S. Chowdhury, et al., "Voice-based navigation system forvisuallyimpaired peopleusing googlett sandgps,"in2020InternationalConference on Computing, Power and Communication Technologies(GUCON), pp. 291–296, IEEE, 2020.
- [12] J. Shen, R. Pang, R. J. Weiss, et al., "Natural tts synthesis by condition-ing wavenet on mel spectrogram predictions," 2018 IEEE InternationalConferenceonAcoustics,SpeechandSignalProcessing(ICASSP),pp.4779–4783,2018.
- [13] M. R. Bhuiyan, M. Rahman, et al., "Ai-powered assistance for thevisuallyimpairedusingobjectdetectionandtext-to-speech,"in20212ndInternational Conference on Robotics, Electrical and Signal ProcessingTechniques (ICREST), pp. 295–300, IEEE, 2021.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)