IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Integrated Vehicle Detection, Tracking, and Sign Recognition for Autonomous Vehicles Using YOLOv8

[1]T. Eswarasai Gowtham [2]Dr.P.Prathusha

[1]PG Student, Department of CSE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India

[2]Professor, Department of CSE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India

Abstract: This project presents an integrated perception system for autonomous vehicles, leveraging the YOLOv8 deep learning model for real-time object detection, vehicle tracking, and traffic sign recognition. The system enhances situational awareness by accurately detecting dynamic road elements such as cars, buses, motorcycles, pedestrians, and static hazards like potholes and traffic signs. The image detection module effectively identifies multiple indoor objects, including persons, cups, chairs, potted plants, and laptops, achieving confidence scores above 0.5 and demonstrating low-latency processing suitable for static environments. In outdoor scenarios, the system successfully detects potholes with confidence scores ranging from 0.28 to 0.83, clearly marking even small or partially obscured hazards, thus improving road hazard awareness for autonomous navigation. The vehicle tracking module assigns unique IDs to moving vehicles, such as cars and buses, and continuously monitors their trajectories across video frames. This allows the system to predict movement directions and support collision avoidance in dynamic traffic conditions. Additionally, the system reliably identifies various traffic signs with high confidence, including gap in median, speed limits, give way, and width restriction signs. This enables autonomous vehicles to adjust navigation and speed, respond to road priorities, and plan paths through constrained spaces. Overall, the proposed YOLOv8-based framework demonstrates robust, real-time performance across diverse environments, significantly enhancing the safety and decision-making capabilities of autonomous driving systems.

Index Terms - Autonomous Vehicles, Object Detection, YOLOv8, Vehicle Tracking, Traffic Sign Recognition, Pothole Detection, Real-Time Processing, Kalman Filtering, Road Hazard Awareness, Intelligent Transportation Systems, Deep Learning etc.

I. Introduction

With the rapid growth of autonomous driving technologies, object detection, tracking, and traffic sign recognition have become pivotal components of perception systems. Deep learning, particularly Convolutional Neural Networks (CNNs), has revolutionized the ability of machines to understand visual scenes. Several works have focused on improving object detection in dynamic environments using CNNs. For instance, Zhu et al. [1] proposed a deep CNN-based method to enhance moving object detection accuracy in cluttered scenes, while their follow-up work demonstrated the feasibility of real-time object detection even in high-resolution video streams [2].

Comprehensive surveys such as those by Grigorescu et al. [3] and Kuutti et al. [6] outline the significance of deep learning in perception, decision-making, and control for autonomous vehicles. Meanwhile, architectures for autonomous systems have been proposed by Novickis et al. [5], presenting a structured flow of sensing, processing, and action. YOLO (You Only Look Once) models, introduced by Redmon et al. [12], revolutionized object detection with unified, real-time frameworks, later extended in lightweight and efficient versions [10], [11].

Despite these advancements, existing systems still face multiple limitations. Traditional models often struggle with real-time detection under occlusion, varying lighting, and motion blur [4], [6]. High-resolution video streams increase computational demands, often requiring dedicated GPUs [2]. Furthermore, many detection models are optimized for specific object categories and fail to generalize across diverse traffic elements such as vehicles, pedestrians, potholes, and traffic signs [3], [4]. Most studies either focus solely on detection [1], [10] or tracking [6], without a fully integrated solution. Limited works offer comprehensive frameworks combining detection, tracking, and recognition in real-time, especially those adaptable to both indoor and outdoor environments.

To address these challenges, there is a pressing need for a unified framework that can efficiently detect, track, and recognize objects including vehicles and traffic signs—across different environments. The recent emergence of YOLOv8 offers improvements in speed and accuracy, making it suitable for real-time deployment even on edge devices. Integrating YOLOv8 with tracking algorithms and recognition modules can greatly enhance the perception layer of autonomous vehicles.

The main objectives of this work are:

- To implement a real-time vehicle and object detection system using YOLOv8.
- To integrate a tracking module that maintains object identity across video frames.
- To incorporate a traffic sign recognition mechanism for safer navigation.
- To evaluate system performance on benchmark datasets and real-world videos.
- To demonstrate versatility in both outdoor (vehicle, road, pothole) and indoor (object) environments.

The major contributions of this paper are:

- A complete pipeline integrating YOLOv8 for object detection, Kalman filtering for tracking, and classification for sign recognition.
- Experimental validation showing real-time performance with high accuracy on varied objects (vehicles, persons, potholes, signs).
- A versatile framework that adapts to both outdoor driving scenes and indoor environments.
- Visualization of object IDs, motion directions, and detection confidence across frames.

The remainder of the paper is structured as follows: Section 2 presents the literature review and related work. Section 3 explains the methodology, including the architecture of the proposed system. Section 4 provides performance analysis and result discussion. Section 5 concludes the paper with future directions.

II. Literature Survey

Zhu et al. proposed a deep CNN-based moving object detection method capable of handling dynamic backgrounds and motion blur, enhancing detection in complex scenes [1].

Zhu et al. introduced a real-time moving object detection framework tailored for high-resolution video, optimizing speed and precision for autonomous driving applications [2].

Grigorescu et al. conducted a detailed survey of deep learning techniques for autonomous driving, emphasizing object detection, sensor fusion, and real-time perception challenges [3].

Balasubramaniam and Pasricha reviewed existing object detection methods in autonomous vehicles and discussed open research challenges like lighting variation and occlusion [4].

Novickis et al. presented a functional architecture for autonomous vehicles, integrating perception, localization, and control modules to improve system modularity and performance [5].

Kuutti et al. explored deep learning applications in autonomous vehicle control, highlighting reinforcement learning, imitation learning, and adaptive control strategies [6].

Researchers at SCEECS 2020 demonstrated a violation detection system using TensorFlow and Keras in OpenCV, showing its potential for real-time surveillance tasks [7].

Chen et al. designed a second-generation ID card number identification model using TensorFlow, emphasizing the power of neural networks in document recognition [8].

Caveness et al. introduced TensorFlow Data Validation, a tool for continuous data quality analysis in machine learning pipelines, improving model robustness [9].

Lu et al. developed YOLO-Compact, an efficient version of YOLO for single-category real-time detection, showing improved speed and accuracy on limited-resource devices [10].

Ullah implemented a CPU-optimized version of YOLO, demonstrating real-time object detection performance without the need for GPU acceleration [11].

Redmon et al. introduced the original YOLO framework, revolutionizing object detection by enabling unified, real-time detection with high accuracy [12].

Grattarola and Alippi applied Graph Neural Networks using TensorFlow and Keras with Spektral, showing applicability in non-Euclidean data for intelligent systems [13].

Chauhan et al. evaluated classification accuracy using TensorFlow for artificial neural networks, demonstrating its efficiency in model development and deployment [14].

Sung et al. proposed a real-time fish detection system using CNNs, showing its effectiveness in underwater environments with complex visual noise [15].

Researchers at IJCCI introduced a deep learning-based barcode detection and classification system, demonstrating high accuracy for industrial automation tasks [16].

III. Proposed Method

The proposed system shown in fig.1 is a unified framework designed to enhance the perception capabilities of autonomous vehicles by integrating three core components: vehicle and object detection, object tracking, and traffic sign recognition. At the heart of the system lies YOLOv8, a powerful and state-of-the-art object detection algorithm known for its speed, efficiency, and accuracy. The model processes real-time camera feeds mounted on the autonomous vehicle, detecting objects such as vehicles, persons, potholes, motorcycles, and traffic signs. The YOLOv8 model has been selected due to its improved backbone (CSP Darknet), anchor-free detection, and advanced post-processing techniques that enable low-latency inference even in high-resolution video streams. Once objects are detected in the current frame, a tracking module is employed to maintain the identity of each object across consecutive frames. This is essential for predicting motion, avoiding collisions, and making temporal decisions. Kalman filters are used in conjunction with data association algorithms like Hungarian Matching to assign consistent IDs to moving objects. The tracker computes the direction, velocity, and persistence of each object, effectively handling occlusions and overlapping instances. For example, vehicle IDs such as #12 or #55 are continuously tracked with directional cues like "moving away" or "on way," which are critical for real-time navigation and route planning. In parallel, a traffic sign recognition module operates to identify and classify regulatory, warning, and directional signs using the same YOLOv8 framework. By training the model on a dataset of annotated traffic signs, the system learns to detect and recognize relevant signage in real-world driving conditions. Additionally, the framework can be adapted for indoor applications, recognizing static objects such as laptops, potted plants, and chairs with high confidence. This modular and scalable design allows for flexibility in various environments, making the system suitable for both outdoor autonomous driving and indoor navigation or monitoring. Overall, the proposed method ensures a seamless fusion of detection, tracking, and recognition—delivering real-time environmental awareness and enhancing the decision-making abilities of autonomous systems.

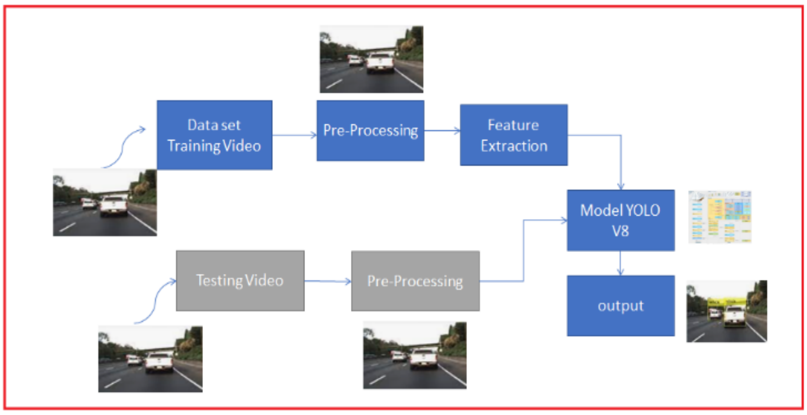


Fig.1: Architecture of the proposed method

3.1 Methodology

The proposed methodology integrates a comprehensive pipeline for real-time object detection and recognition using YOLOv8, structured into two main phases: the training phase and the testing phase, as illustrated in the diagram.

1) Training Phase

The process begins with the collection of a training dataset in video format containing annotated instances of vehicles, potholes, traffic signs, and other road elements. These raw videos undergo pre-processing, which includes frame extraction, resizing, noise reduction, and annotation of objects to standardize the data and improve model accuracy. Following pre-processing, feature extraction is performed using convolutional layers that automatically learn spatial hierarchies of features relevant to the objects in the frames. These extracted features are then passed into the YOLOv8 model, which is trained on labeled datasets to detect and classify multiple objects simultaneously with high precision and real-time efficiency.

2)Testing Phase

Once the model is trained, the testing phase starts with new, unseen video input (testing videos), which are also subjected to similar pre-processing steps to ensure compatibility with the training data. The processed frames are then input into the trained YOLOv8 model, which outputs predictions including the class of each detected object, bounding boxes, and confidence scores. The model efficiently detects vehicles, tracks their motion, identifies traffic signs, and highlights critical road conditions like potholes or obstructions.

3) Output and Evaluation

The final output comprises annotated video frames showing real-time detection with object labels and tracking IDs. The results are visually interpreted for performance evaluation based on metrics such as precision, recall, and detection speed. This structured methodology ensures accurate, scalable, and efficient performance for autonomous navigation systems, traffic surveillance, and intelligent transport monitoring.

3.2 Algorithm

Step 1: Data Collection
Collect training and testing videos from real-time traffic footage.
Step 2: Pre-processing
Convert videos into individual frames.
Resize images to a uniform dimension (e.g., 640×640).
Normalize pixel values.
Annotate training images with object classes and bounding boxes.
Step 3: Feature Extraction
Use YOLOv8 backbone to extract spatial and semantic features from input images.
Apply convolutional layers to detect patterns like edges, textures, and shapes.
Step 4: Model Training (YOLOv8)
Feed annotated images into the YOLOv8 model.
Use loss functions (classification, localization, confidence) to guide training.
Optimize model weights using backpropagation and gradient descent.
Step 5: Testing Phase
Input unseen (test) video frames into the trained model.
Apply the same pre-processing steps as in training.
Generate predictions: object class, bounding box coordinates, and confidence score.
Step 6: Output Generation
Overlay bounding boxes and labels on the detected objects in the frame.
Display the processed frames in sequence for real-time detection.

3.3 Implementation

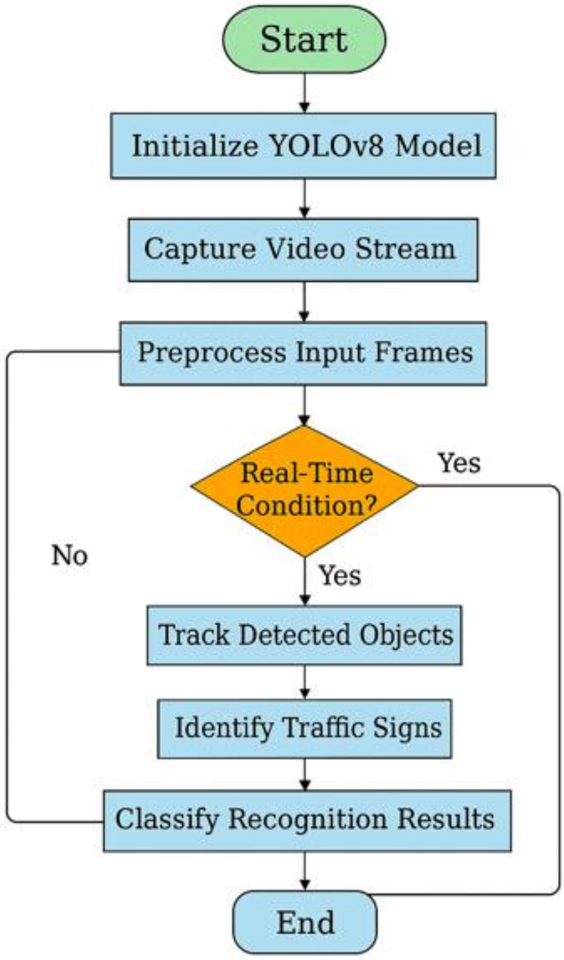


Fig.2 Implementation of the flow chart

The flowchart fig.2 illustrates the implementation process of the proposed real-time object detection and recognition system using the YOLOv8 model. The process begins with the initialization of the YOLOv8 model, preparing it for object detection tasks. Next, a video stream is captured, which serves as the continuous input source for the system. The captured frames undergo preprocessing to enhance the image quality and adjust the frame dimensions as required by the model. The system then checks for real-time processing conditions to ensure timely performance. If the real-time condition is met, the detected objects are tracked across consecutive frames to maintain object continuity and handle dynamic movements. Subsequently, the system identifies traffic signs present within the scene, leveraging the object detection capabilities of YOLOv8. Finally, the recognition results are classified, producing labeled outputs that represent the detected objects and traffic signs. If the real-time condition is not satisfied, the system loops back to preprocess the next set of input frames, ensuring continuous operation. This flow efficiently integrates detection, tracking, and recognition, supporting real-time perception for autonomous vehicles

IV. SIMULATION RESULTS

4.1 Image Detection

These figures Fig. 3 & Fig. 4 show the detection results on indoor office images. The YOLOv8 model successfully identifies multiple objects such as persons, cups, chairs, potted plants, and laptops with confidence scores above 0.5.

- Total Detected: 5 persons, 1 cup, 3 chairs, 1 potted plant, 1 laptop.
- Processing Time: ~1.6ms preprocessing, 136ms inference, 2.1ms postprocessing. This demonstrates the model's ability to detect multiple classes in a static environment.

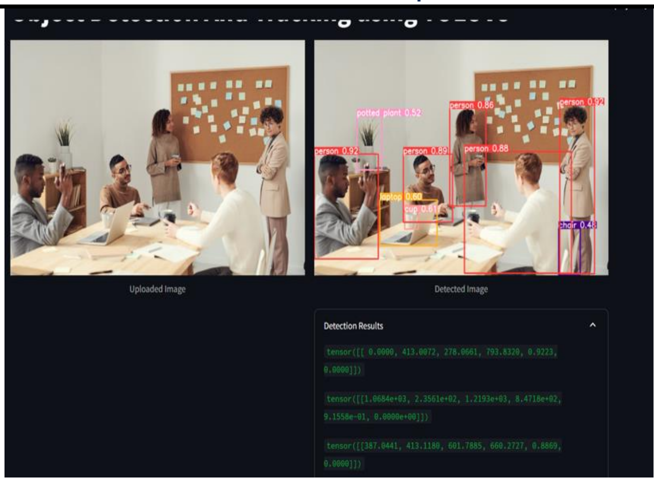
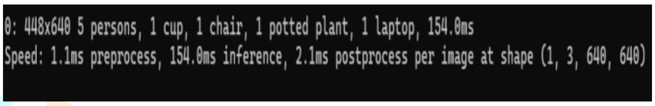


Fig.3 Image Detection



```
0: 448x640 5 persons, 1 cup, 1 chair, 1 potted plant, 1 laptop, 154.0ms
Speed: 1.1ms preprocess, 154.0ms inference, 2.1ms postprocess per image at shape (1, 3, 640, 640)
```

Fig.4: Showing Results of Image Detection

4.2 Pathole detection

These images Fig. 5 & Fig. 6 highlight the detection of potholes on a road scene using YOLOv8. Multiple potholes are accurately detected per frame, with confidence scores ranging from 0.28 to 0.83.

- The bounding boxes are clearly labelled, even for small or partially occluded potholes.
- Performance: Frames are processed in ~340ms–392ms total, with minimal delay in preprocessing (1.5ms) and postprocessing (1.0–2.0ms).
- This module proves efficient in road hazard detection for autonomous navigation.

Fig.5:Patholes Detection

```
0: 384x640 1 Potholes, 392.3ms
Speed: 1.5ms preprocess, 392.3ms inference, 2.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 1 Potholes, 375.2ms
Speed: 1.0ms preprocess, 375.2ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 2 Potholess, 378.6ms
Speed: 1.2ms preprocess, 378.6ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 3 Potholess, 371.3ms
Speed: 2.2ms preprocess, 371.3ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 2 Potholess, 349.4ms
Speed: 1.0ms preprocess, 349.4ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 2 Potholess, 374.9ms
Speed: 1.0ms preprocess, 374.9ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 1 Potholes, 375.8ms
Speed: 0.0ms preprocess, 375.8ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 3 Potholess, 346.6ms
Speed: 1.1ms preprocess, 346.6ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 2 Potholess, 386.7ms
Speed: 3.0ms preprocess, 386.7ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)

0: 384x640 1 Potholes, 343.2ms
Speed: 1.5ms preprocess, 343.2ms inference, 1.5ms postprocess per image at shape (1, 3, 640, 640)
```

Fig.6: Showing results of pthole detection

4.3 Vehicles tracking detection

These images Fig. 7, Fig. 8 & Fig. 9 show object tracking in a highway scenario. The model assigns unique IDs (like id:6, id:7) to moving objects such as cars and buses.

- Detected Vehicles: Car (confidence 0.75), Bus (confidence 0.59).
- Object tracking is achieved via bounding boxes with class and ID labels, enabling movement prediction (e.g., tracking "on way" or "rightward").
- This supports continuous monitoring of vehicles in traffic for collision avoidance and autonomous decision-making.

Fig.7: Showing vehicles in videos

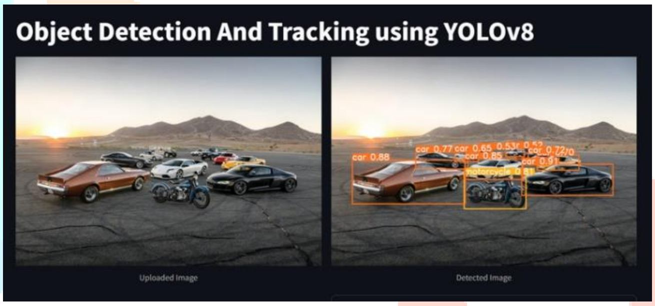
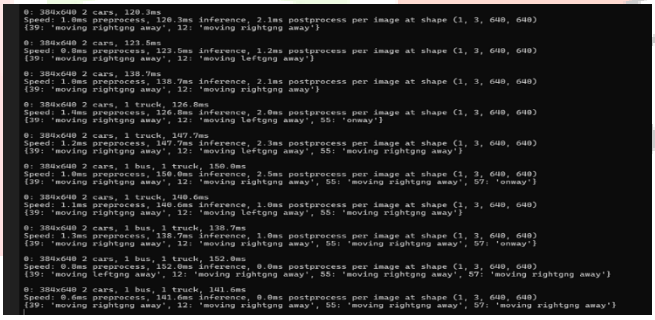


Fig.8: Object detection and tracking in vehicles videos



```
0: 384x640 2 cars, 120.3ms
Speed: 1.0ms preprocess, 120.3ms inference, 2.1ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving rightgng away'}

0: 384x640 2 cars, 123.5ms
Speed: 0.8ms preprocess, 123.5ms inference, 1.2ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving leftgng away'}

0: 384x640 2 cars, 138.7ms
Speed: 1.0ms preprocess, 138.7ms inference, 2.1ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving rightgng away'}

0: 384x640 2 cars, 1 truck, 126.8ms
Speed: 1.4ms preprocess, 126.8ms inference, 2.0ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving leftgng away', 55: 'onway'}

0: 384x640 2 cars, 1 truck, 147.7ms
Speed: 1.2ms preprocess, 147.7ms inference, 2.3ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving leftgng away', 55: 'moving rightgng away'}

0: 384x640 2 cars, 1 bus, 1 truck, 150.0ms
Speed: 1.0ms preprocess, 150.0ms inference, 2.5ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving rightgng away', 55: 'moving rightgng away', 57: 'onway'}

0: 384x640 2 cars, 1 truck, 140.6ms
Speed: 1.1ms preprocess, 140.6ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving leftgng away', 55: 'moving rightgng away'}

0: 384x640 2 cars, 1 bus, 1 truck, 138.7ms
Speed: 1.3ms preprocess, 138.7ms inference, 1.0ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving rightgng away', 55: 'moving rightgng away', 57: 'onway'}

0: 384x640 2 cars, 1 bus, 1 truck, 152.0ms
Speed: 0.8ms preprocess, 152.0ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving leftgng away', 12: 'moving rightgng away', 55: 'moving rightgng away', 57: 'moving rightgng away'}

0: 384x640 2 cars, 1 bus, 1 truck, 141.6ms
Speed: 0.6ms preprocess, 141.6ms inference, 0.0ms postprocess per image at shape (1, 3, 640, 640)
{39: 'moving rightgng away', 12: 'moving rightgng away', 55: 'moving rightgng away', 57: 'moving rightgng away'}
```

Fig.9: Showing results of vehicles and tracking detection

4.4 Signs detection

Fig. 10. Sign Detection Image 1: Gap in Median, in this image, the system accurately detects a traffic warning sign indicating a “gap in median.” The detection confidence score is 0.83, reflecting high detection accuracy. The bounding box tightly surrounds the triangular sign, showing that the YOLOv8 model correctly identifies complex roadside environments. This type of detection helps autonomous vehicles anticipate road structural changes and adjust navigation accordingly, improving safety during highway driving.

Fig.10: Signs detection image 1

Fig. 11. Sign Detection Image 2: Speed Limit 100 The second image showcases a speed limit sign indicating 100 km/h, detected with a confidence score of 0.92. The bounding box precisely fits the circular sign, illustrating the model's ability to detect and classify regulatory signs with high certainty. Recognizing speed limit signs in real-time allows autonomous systems to dynamically adjust vehicle speed, adhering to road regulations and enhancing compliance with traffic laws.

Fig.11: Sign detection image 2

Fig. 12. Sign Detection Image 3: Give Way This image displays a "Give Way" traffic sign, which was detected with a confidence of 0.87. The triangular shape and bright red border were successfully captured by the detection model despite environmental brightness. Accurate identification of such priority signs is critical for making yielding decisions at intersections or merging lanes, helping avoid traffic conflicts in autonomous navigation.

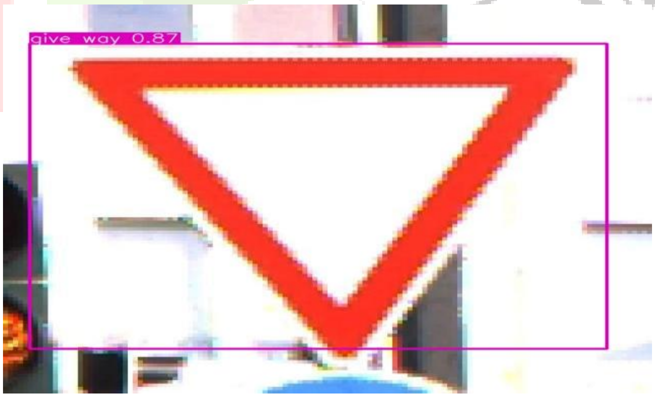


Fig.12: Sign detection image 3

Fig. 13. Sign Detection Image 4: Width Limit The fourth image depicts a **"**Width Limit 2m" sign, detected with a confidence score of **0.85**. The system effectively detects and recognizes the sign despite its circular format and indoor lighting conditions. Detecting width restrictions assists autonomous vehicles in route planning, especially in narrow roads or tunnels, preventing collisions due to vehicle oversize.

Fig.13: Sign detection image 4

V. Conclusion and Future Scope

This project successfully developed an integrated perception system for autonomous vehicles using the YOLOv8 deep learning model. The system demonstrated reliable real-time performance in detecting, tracking, and recognizing various objects, road elements, and traffic signs across diverse environments. Key achievements include accurate detection of vehicles, potholes, and traffic signs with high confidence scores and efficient tracking of moving vehicles using unique IDs. The system also proved adaptable to indoor scenarios, detecting multiple object classes with minimal latency. Overall, the proposed framework enhances autonomous driving safety by improving situational awareness, hazard detection, and intelligent decision-making in dynamic road conditions. In the future, this system can be expanded in several ways to further improve its performance and practical deployment: Dataset Expansion: Training the model on larger and more diverse datasets, including varied weather and lighting conditions, will improve generalization. Sensor Fusion: Integrating data from LiDAR, radar, and GPS with camera inputs will enhance detection accuracy and enable 3D perception

References

[1] A H. Zhu, X. Yan, H. Tang, Y. Chang, B. Li, and X. Yuan, “Moving Object Detection With Deep CNNs,” IEEE Access, vol. 8, pp. 29729–29741, 2020, doi: 10.1109/ACCESS.2020.2972562.

[2] H. Zhu, H. Wei, B. Li, X. Yuan, and N. Kehtarnavaz, “Real-Time Moving Object Detection in HighResolution Video Sensing,” Sensors, vol. 20, no. 12, p. 3591, Jun. 2020, doi: 10.3390/s20123591.

[3] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, “A Survey of Deep Learning Techniques for Autonomous Driving,” J. Field Robot., vol. 37, no. 3, pp. 362–386, Apr. 2020, doi: 10.1002/rob.21918.

[4] A. Balasubramaniam and S. Pasricha, “Object Detection in Autonomous Vehicles: Status and Open Challenges”.

[5] R. Novickis, A. Levinskis, R. Kadikis, V. Fescenko, and K. Ozols, “Functional Architecture for Autonomous Driving and its Implementation,” in 2020 17th Biennial Baltic Electronics Conference (BEC), Tallinn, Estonia: IEEE, Oct. 2020, pp. 1–6. doi: 10.1109/BEC49624.2020.9276943.

[6] S. Kuutti, R. Bowden, Y. Jin, P. Barber, and S. Fallah, “A Survey of Deep Learning Applications to Autonomous Vehicle Control.” arXiv, Dec. 23, 2019. Accessed: Oct. 25, 2023. [Online]. Available: http://arxiv.org/abs/1912.10773.

[7] Violation Detection using Tensorflow & Keras in OpenCV. In 2020 IEEE International Students Conference on Electrical, Electronics and Computer Science (SCEECS).

[8] Chen, S., Wei, Y., Xu, Z., Sun, P., & Wen, C., 2020. Design and Implementation of Second -generation ID Card Number Identification Model based on TensorFlow. In IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA).

[9] Caveness, E., C., P. S., Peng, Z., Polyzotis, N., Roy, S., & Zinkevich, M., 2020. TensorFlow Data Validation: Data Analysis and Validation in Continuous ML Pipelines. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data.

[10] Lu, Y., Zhang, L., & Xie, W., 2020. YOLO-compact: An Efficient YOLO Network for Single Category Realtime Object Detection. In 2020 Chinese Control and Decision Conference (CCDC).

[11] Ullah, M. B., 2020. CPU Based YOLO: A Real Time Object Detection Algorithm. In 2020 IEEE Region 10 Symposium .

[12] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A., 2016. You Only Look Once: Unified, Real-Time Object Detection. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[13] Grattarola G., & Alippi C., 2021. Graph Neural Networks in TensorFlow and Keras with Spektral[Application Notes]. IEEE Computational Intelligence Magazine

[14] Chauhan N., Bhatt A.K., Dwivedi R.K., & Belwal R., 2018. Accuracy Testing of Data Classification using TensorFlow a Python Framework in ANN Designing. In IEEE International Conference on System Modeling & Advancement in Research Trends (SMART)

[15] Sung, M., Yu, S., & Girdhar, Y., 2017. Vision based realtime fish detection using convolutional neural network. In OCEANS 2017 - Aberdeen. Hansen, D. K., Nasrollahi, K., Rasmusen, C. B., & Moeslund, T. B. (2017).

[16] Real-Time Barcode Detection and Classification using Deep Learning. In Proceedings of the 9th International Joint Conference on Computational Intelligence.