Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 15s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

YOLOv8m: A Deep Learning Approach to Traffic Sign Recognition in CARLA and Speed control of Autonomous Vehicles

Yamini Tondepu[1], P Manivannan[*2], PR. Sathappan[3], S. Harikishore[4], Viswanathan Ramasamy Reddy[5], Dr. T. Vengatesh[6]

[1]Assistant professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur – 522 303, Andhra Pradesh.

[*2]School of Computing, SASTRA University,

[3,4]Student , School of mechanical, SASTRA University,

[5]Professor ,Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur – 522 303, Andhra Pradesh.

[6]Assistant Professor, Department of Computer Science, Government Arts and Science College, Veerapandi, Theni, Tamilnadu, India.

[1]Email ID: smile.yamini@gmail.com,[*2]Email ID: gotomanivannan@yahoo.in, [3] Email ID: sathappanpr2002@gmail.com,

[4] Email ID: kishoresundar2003@gmail.com, [5] Email ID: rvnathan06@gmail.com, [6] Email ID: venkibiotinix@gmail.com

***Corresponding Author:**

P Manivannan

School of Computing, SASTRA University,

Email ID: gotomanivannan@yahoo.in

Cite this paper as: Yamini Tondepu, P Manivannan, PR. Sathappan, S. Harikishore, Viswanathan Ramasamy Reddy, Dr. T. Vengatesh, (2025) YOLOv8m: A Deep Learning Approach to Traffic Sign Recognition in CARLA and Speed control of Autonomous Vehicles. *Journal of Neonatal Surgery*, 14 (15s), 61-76.

ABSTRACT

Traffic sign detection acts as the eyes for autonomous vehicles, employing computer vision to decipher road signs for safe and compliant navigation. This technology tackles challenges like variable lighting and occlusions by leveraging deep learning models trained on diverse datasets. By interpreting signs accurately, autonomous vehicles can navigate roads confidently, paving the way for a safer future. Traffic sign detection in real-world scenarios faces a confluence of challenges: varying lighting and weather conditions can degrade sign visibility, partial or complete occlusions by other objects can hinder detection, and the sheer diversity of traffic signs across regions necessitates robust models capable of generalizability. The research utilizes a combined real-world (65%) and simulated (CARLA, 35%) dataset for training a YOLOv8m model for speed limit sign detection (30, 60, 90 km/h) in autonomous vehicles. Image augmentation and collaborative annotation via Makesense platform enrich the dataset. The model has achieved 98% accuracy rate in detecting speed limits on the road side, even in difficult situations. The model is evaluated in the CARLA simulator for controlled testing before real-world implementation.

Keywords: *Traffic sign detection, occlusions, Makesense platform, Autonomous vehicles, Computer version, YOLOv8m model, CARLA.*

1. INTRODUCTION

Domains such as image and video analysis have seen significant growth in application over the past few years. The two primary technologies dictating technical society are Computer Version and Artificial intelligence. Over years, human intelligence has been trained to recognize and comprehend scenes that are captured by the eyes. Traffic control is one of the most significant subjects in the transportation industry. Governments can enhance the quality of a road by using information about the vehicles that are traveling on it. Autonomous vehicles are engineered to maximize traffic flow, reduce the frequency of traffic incidents, enable human transportation of goods in hazardous environments, increase accessibility for individuals with disabilities, and more.

Throughout history, vision-based technology has evolved from being merely a sensing modality to intelligent computing systems capable of comprehending the physical world. Object detection and tracking are major challenges for computer vision applications such as surveillance, autonomous robot navigation, and vehicle navigation. Vehicles and other real-world objects can be tracked in a dynamic environment using video surveillance. A self-driving car makes decisions about how to control its speed, direction, and acceleration by continuously absorbing data from multiple onboard sensors, including RADAR, LIDAR, and a camera. These sensors help the car identify obstacles and lanes on the road. Obstacle detection methods such as LIDAR and RADAR have proven to be accurate. The CARLA simulator, an open-source simulator for research on autonomous driving, has made R&D in this area more accessible in recent years[1]. Designed from the ground up, CARLA is intended to facilitate the research, development, testing, and certification of autonomous urban driving systems. [2] This paper proposes a new obstacle detection system for drones using multiple ultrasonic sensors and signal processing techniques. This offers an alternative to high-power lasers that could blind pilots and has better resolution than standard ultrasonic sensors.

The remainder of this essay is divided into several sections. The literature reviews and problem analysis are briefly discussed in Section II. Section III introduces CARLA (Car Learning to Act), providing essential context for our work and advantages. The proposed method is covered in Section IV. Common evaluation metrics and performance are explained in Section V. Concluding remarks are provided in Section VI.

2. LITERATURE REVIEW

In this research [3] they proposed a new method for detecting red and blue traffic signs in color images. It uses a combination of color and texture features with a Quadratic SVM classifier and achieves high accuracy (98.5%) on a benchmark dataset. The research doesn't explore using deep learning techniques, which are becoming increasingly popular and powerful for object detection tasks. They could also improve their method by incorporating shape features or considering signs beyond red and blue. The [4] paper surveys traffic sign detection (TSD) for driver assistance systems. It highlights the need for standardized datasets and explores open challenges like context integration and non-European signs. The paper doesn't discuss the potential of deep learning for TSD, a rapidly advancing field. They could explore how to combine TSD with driver behavior analysis for a more comprehensive driver assistance system.

This study [5] examines the state-of-the-art in deep learning-based object detection, including new developments, applications, benchmark datasets, and future approaches. It gives a thorough rundown of the subject. This [6] study investigates the benefits of edge computing for computer vision applications and the Internet of Things (IoT). It covers the essential ideas, hardware, benefits, and drawbacks of this developing industry. The use of edge computing in practical Internet of Things and computer vision applications is not discussed in detail in this study. Furthermore, it ignores any security issues that can arise from processing data at the edge. This research investigates [7] the use of YOLOv3 for multiple object detection (5 classes) with vehicle tracking in traffic videos, and Convolutional Neural Networks (CNNs) for single object detection (3 classes) under different lighting conditions. The performance comparison of their single object detection CNN with other current models is not discussed in the publication. By employing more advanced techniques than only centroid tracking, they might enhance tracking.

The Single Shot Detector (SSD) algorithm [8] is used in this paper's deep learning model to identify and pinpoint human activity in videos for security purposes. Although the model's accuracy (precision: 0.775, recall: 0.679) is good, it still has to be improved before being used in real time. The performance of their model is not compared to other methods that are currently in use for human activity recognition in the paper. They don't go into enough detail on how to deal with issues like real-time processing and data availability for useful applications. [9] The significance of network depth was demonstrated when a deep CNN with 60 million parameters was trained to categorize images in the ImageNet competition, obtaining top-1 and top-5 error rates of 37.5% and 17.0%, respectively. In order to enhance performance, the paper could investigate unsupervised pre-training and take into account the computational difficulties involved in scaling the network to handle video sequences.

This study offers [10] a fresh method for understanding quantum convolutional neural networks (QCNNs) and how they might be used for object and image detection. While acknowledging the difficulties of limited access to quantum computers and appropriate training algorithms, the authors contend that QCNNs may perform better than classical approaches. The performance of QCNN and classical CNN on the same datasets is not directly compared in the research. It also doesn't include a critical analysis of how much energy QCNNs use when operating in comparison to traditional techniques. By mapping [11] camera pixels to steering signals, the CNN-based system learns to drive independently and can operate well in a variety of settings without the need for explicit feature labeling. The decision-making process of the system under erratic situations and the moral implications of AI-driven cars in practical settings may be covered in the article.

Using data [12] from various sensors, the study focuses on 3D object detection and localization for autonomous driving. It discovers that models trained on synthetic data perform poorly on real-world data. The adaptation of models trained on synthetic data to real-world events and the possibility of using transfer learning to close this gap might have been examined in the article. This study examines deep learning techniques for autonomous driving behavior prediction in nearby vehicles.

Although these methods increase hazard awareness, they have drawbacks. Specifically, they don't take into account the surroundings, traffic laws, sensor limits, or computing expenses. The review did not address how these deep learning techniques address real-world autonomous driving scenarios with respect to weather, traffic laws, sensor limitations, and computational constraints.

Tools like the CARLA simulator, which support object detection algorithm training, are helping to advance research on autonomous vehicles [13]. Despite difficulties, important technologies like RNN and QCNN are being applied, and progress is noteworthy—using the SSD Model, 82.81% accuracy was attained. The ethical ramifications, legal difficulties, and effects on employment in the driving industry are not included, nor are they addressed by them. Furthermore, a thorough examination of the data privacy issues surrounding autonomous car technology is absent. In terms [14] of vehicle trajectory prediction, the Memory Neuron Network (MNN) is a novel RNN that outperforms state-of-the-art models with less computing effort and complexity. It has demonstrated improved results when validated on both the NGSIM dataset and a dataset created using the CARLA simulator.

Users can experience the behaviour of a self-driving car in advance via Auto Preview, which increases user confidence and comprehension [15]. They emphasize immediate gains at the expense of investigating long-term impacts on consumer reliance. The long-term effects of employing Auto Preview on user trust and system dependence were not investigated in this study. This study presents [16] an overview of different object tracking algorithms for video surveillance, emphasizing the significance of feature selection and suggesting a single technique to improve tracking precision. A more thorough examination of the algorithms' computational effectiveness and scalability to other surveillance settings would be beneficial for the discussion. [17] Traffic systems' object detection and tracking have greatly improved thanks to AI and CV technologies, especially CNN and YOLOv3 models, which achieve high accuracy and real-time performance. The study might investigate how different environmental factors affect model performance and think about incorporating more datasets for a more thorough analysis.

The use of temporal detection and background removal techniques for accurate object detection in video surveillance is covered in the study[18]. By monitoring different regions independently, these techniques improve processing speed and accuracy. The integration of machine learning for predictive analytics and the effects of weather and dynamic lighting on detection accuracy could be further explored in this article. In order to detect and recognize traffic signs in India, a new deep learning model (RMR-CNN) is proposed in this study[19] . When used to a bespoke dataset of Indian traffic signs taken under different circumstances, it obtains remarkable accuracy (precision: 97.08%). How well RMR-CNN performs on benchmark datasets in comparison to other cutting edge traffic sign recognition algorithms is not mentioned in the research. By including techniques to deal with soiled or cleaned traffic signs and larger viewing angles, they could increase robustness.

With plans for on-road testing, the article provides a CNN-based strategy for a level-2 autonomous car that makes use of sensors and simulators for real-time steering and obstacle avoidance [20]. The study may explore the moral ramifications of self-driving cars as well as the necessity of rigorous field testing to guarantee dependability and safety in a variety of scenarios. 95% accuracy was attained using an AI-based CNN-based traffic sign recognition system that processed data in real time on a CPU [21]. It emphasizes how resilient the system is to zooms and rotations of images. The system's functionality in bad weather and how it works with other ADAS elements to provide complete driver assistance might be covered in the article. [22] The system uses a CNN model that is TensorFlow-implemented to detect and recognize traffic signs with high accuracy, which is essential for road safety. It could be improved by adding non-circular sign detection and strengthening resistance to outside influences like low visibility.

In order to boost frame rate to 32 frames per second (FPS) without the need for a GPU, this article suggests a color and form segmentation technique to lessen the workload of AI for traffic sign detection in autonomous vehicles[23]. The system's performance with non-standard traffic signs and in inclement weather is not discussed in the study. To increase frame rate even higher without compromising accuracy, they could possibly investigate more complex filtering strategies. The article provides a Faster R-CNN with Inception Resnet V2 traffic sign identification system that is improved by a novel blurring preprocessing to improve detection[24]. Although it displays encouraging outcomes, the classification stage does not make use of category information from detection. By adding the category data from detection to the classification process, the study could improve the system even more and possibly increase performance and accuracy.

This work [25] presents the Iranian Vehicle Volume Database (IRVD), a sizable dataset of Iranian cars taken in a range of scenarios for license plate identification and vehicle categorization. They provide a lightweight CNN architecture that can analyze data in real time and achieves high accuracy (99.09%). To illustrate IRVD's comprehensiveness, the paper does not contrast it with other vehicle datasets currently in existence. By incorporating more modern and sophisticated deep learning models for classification, they could enhance the evaluation. Using a unique traffic image dataset, the YOLOv4 object detection model is trained to identify nine different vehicle types[26] . The real-time object detection algorithm that combines classification and bounding box prediction into a single neural network. Unlike multi-step approaches, YOLO processes the entire image at once, dividing it into regions and predicting bounding boxes and probabilities for each region. This efficient architecture strikes a balance between speed and accuracy, making it suitable for applications like robotics, driverless cars, and video surveillance. The YOLO family has evolved through iterations (from YOLOv1 to YOLOv8),

with improvements in network design, loss functions, anchor boxes, and input resolution scaling as shown in Figure 1. For unseen traffic photos, the model demonstrated high sensitivity (94.44%) and accuracy (99.28%). The testing on real-world traffic videos, which is essential for real-world application, is not mentioned in the study. The performance of their model is not contrasted with that of other object detectors using the same dataset.

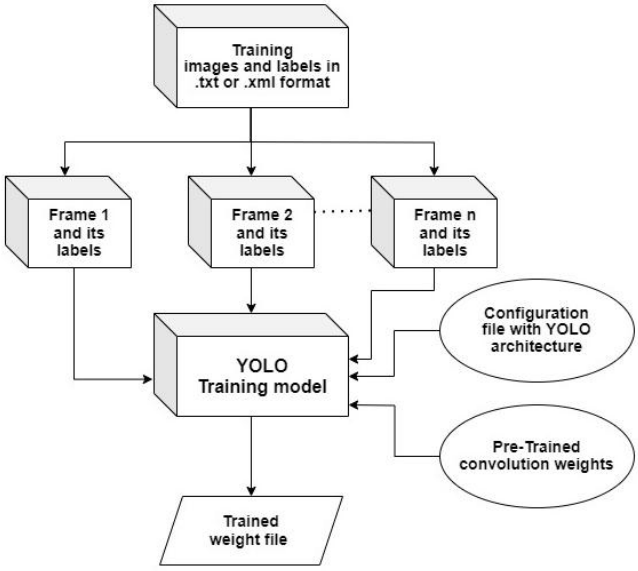

Figure 1: Work flow of YOLO models

In the upcoming sections, paper delve into the intricacies of our research. Section II introduces CARLA (Car Learning to Act), providing essential context for our work and advantages. Section III outlines our Proposed Work, detailing the novel approach we've taken. Moving forward, Section IV delves into the Experimental results, shedding light on the methodology employed and webcam results. Finally, Section V concludes our exploration

3. CARLA (CAR LEARNING TO ACT)

CARLA is an open-source autonomous driving simulator designed from the ground up to support the development, training, and validation of autonomous driving systems. Its modular and flexible API addresses a wide range of tasks related to autonomous driving. One of the primary goals of CARLA is to democratize autonomous driving research and development, making it accessible and customizable for users. The platform provides not only open-source code and protocols but also open digital assets, including urban layouts, buildings, and vehicles, specifically created for autonomous driving research. Researchers can freely utilize these assets to create realistic scenarios. CARLA allows flexible specification of sensor suites, environmental conditions, and full control over static and dynamic actors. Additionally, it supports map generation, making it a powerful tool for testing and validating autonomous driving algorithms

CARLA employs a server multi-client architecture, enabling multiple clients to control different actors within the same simulation or across different nodes. This scalability facilitates collaborative research and testing, allowing researchers to simulate complex scenarios involving various vehicles, pedestrians, and environmental conditions. Researchers can leverage CARLA's flexible API to control all aspects of the simulation. From traffic generation and pedestrian behaviours to weather conditions and sensor configurations (including LIDARs, cameras, depth sensors, and GPS), CARLA provides fine-grained control. This versatility makes it an ideal platform for testing and benchmarking autonomous driving algorithms and sensor fusion techniques

The implementation block diagram is shown in Figure 2. In this work, an object detection model was trained using a dataset that was first generated using the CARLA simulator. Later, test photos were used to evaluate the model's performance on CARLA.

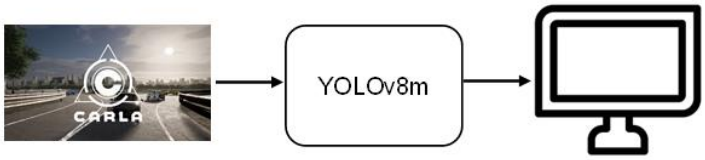

Figure 2: Block diagram of implementation

4. PROPOSED WORK

The dataset leverages a combination of simulated and real-world images. Approximately 65% of the images are sourced from the real world, while the remaining 35% are captured within the CARLA simulator Figure 3, a platform for developing autonomous vehicle applications.

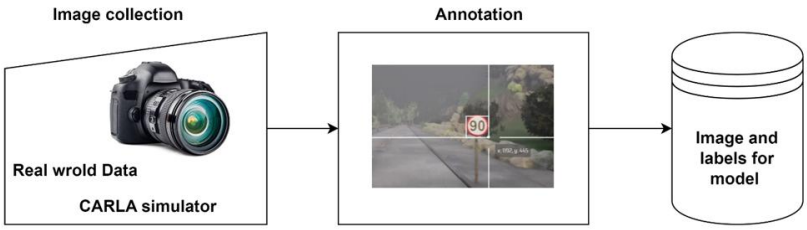

Figure 3: Dataset collection and annotations of the images

To enrich the dataset and improve model generalizability, image augmentation techniques are employed. This process involves manipulating the existing images to generate variations, such as rotations, flips, and color jittering. This expands the dataset size and introduces the model to a wider range of image characteristics, potentially enhancing its robustness and performance. The annotation process, likely involving labeling image elements or creating segmentation masks, is conducted using the Makesense platform. This online platform facilitates collaborative annotation tasks, allowing human annotators to efficiently label the augmented images. Following the data acquisition and augmentation steps, the annotated images are used to train a YOLOv8m model. This indicates a pre-trained YOLOv8 model variant, likely the "medium" version designed for a balance between accuracy and speed Figure 4. 80 epochs, refers to the number of times the entire training dataset is passed through the model for learning. Here, 80 epochs are specified. This defines the batch size 15, which is the number of images shown to the model during each training iteration.

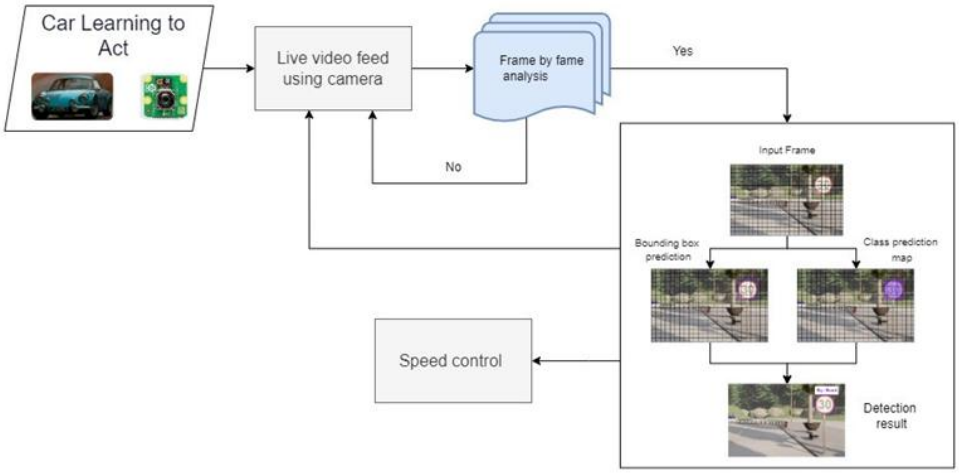

Figure 4: Proposed method work flow

A batch size of 15 is used in this configuration. Training an object detection model involves feeding it annotated images

where objects are labelled with bounding boxes or other relevant information. YOLOv8, the successor to the popular YOLO object detection system, is expected to make significant strides in accuracy and efficiency. It might leverage cutting-edge backbone architectures to extract richer visual features, enabling more precise object detection across various scenarios. Furthermore, YOLOv8 is likely to prioritize real-time performance through efficient model design techniques. The possibility of YOLOv8 expanding beyond just detection to encompass tasks like segmentation or pose recognition is also intriguing. Additionally, the research might explore tailoring the model for specific domains through customized training and potentially even integrating it with reinforcement learning for intelligent decision-making. These advancements position YOLOv8 to be a powerful tool for various object detection applications. The "m" variant suggests a medium-sized model offering a balance between these aspects. Epochs control the number of times the model sees the entire dataset. Here, 80 epochs allow for extensive training. Batch size determines the number of images processed at once. A smaller batch size (15) might be suitable for resource constraints or fine-tuning the model.

The research leverages the CARLA simulator, a software platform designed to develop and evaluate applications for autonomous vehicles in a virtual environment. Within this simulated world, the YOLOv8m model, trained as described earlier, is deployed to detect specific objects: speed limit signs (30, 60, and 90 km/h). The YOLOv8m model continuously processes frames captured from the CARLA simulator. When the model detects a speed limit sign within an image frame, it likely outputs the identified speed limit value (e.g., 30 km/h). The first frame will indicate the speed of the car, while second frame indicates the object detection of the model in the figure 5. Initial speed of the car is 33km/h, when a speed limit of 30 km/h is detected, as indicated in Figure 5, the vehicle's onboard system responds promptly. After crossing the speed limit sign, as depicted in Figure 6, the car automatically adjusts its speed to 23km/h, which is below 30 km/h. The speed will increase or decrease according to the next signboard.

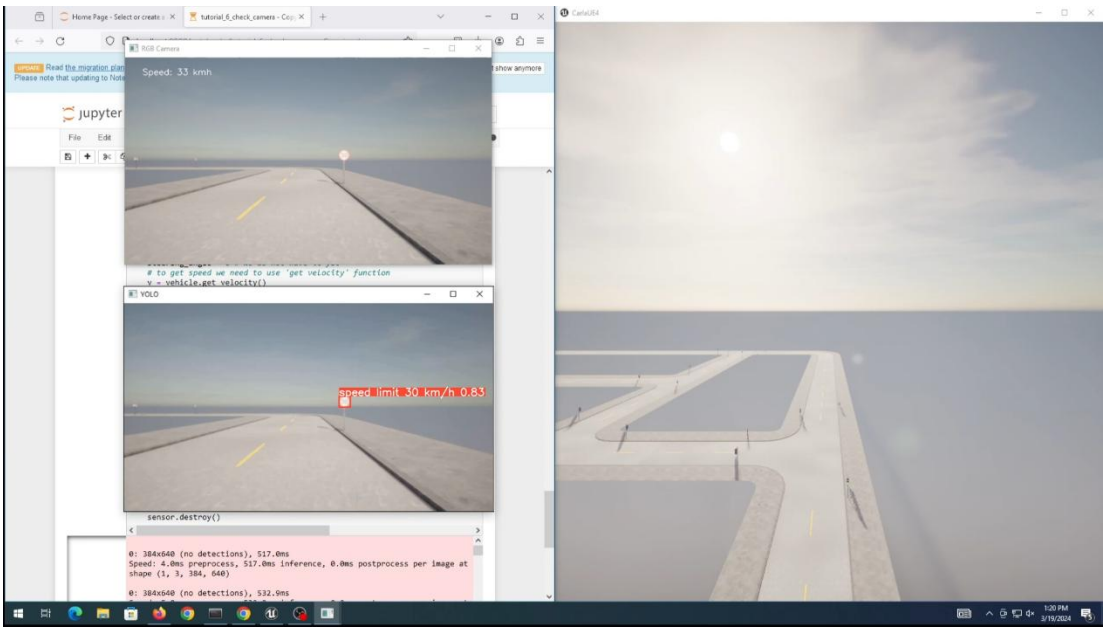

Figure 5: Detecting the sign board

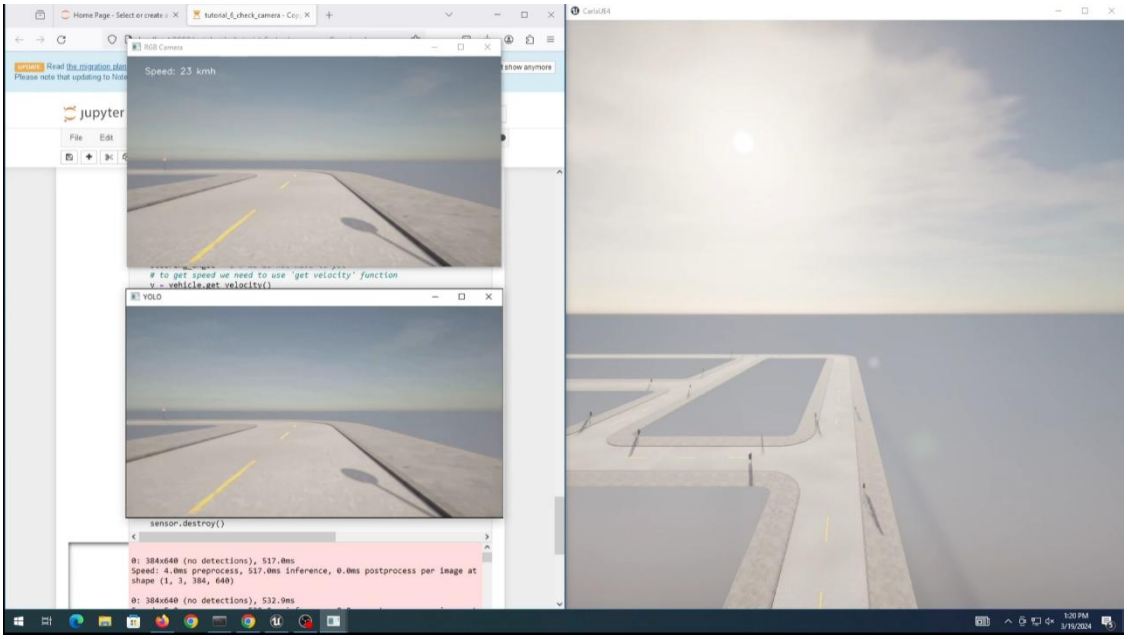

Figure 6: Controlling the speed according to the sign board

This approach offers a controlled environment to evaluate the effectiveness of the YOLOv8m model for speed limit sign detection in autonomous vehicle applications. It allows for testing the model's performance under various simulated scenarios and lighting conditions without the risks of real-world testing. Successful implementation paves the way for integrating the model into real autonomous vehicles for traffic sign recognition and potentially for adhering to speed limits. The research leverages the CARLA simulator, a software platform designed to develop and evaluate applications for autonomous vehicles in a virtual environment. Within this simulated world, the YOLOv8m model, trained as described earlier, is deployed to detect specific objects: speed limit signs (**currently focusing on 30, 60, and 90 km/h**). This research serves as a foundation for a more comprehensive speed limit sign detection system.

5. EXPERIMENTAL SETTINGS

This is probably referring to the bounding box regression loss function used in object detection tasks Usually, "Box" stands for the bounding boxes that are estimated to surround an object in an image. This is a typical abbreviation for "classification loss" in tasks involving object identification and categorization. It calculates the difference between ground truth labels and expected class probabilities. Distribution focal loss(dfl), this loss function is frequently applied to jobs involving object detection, especially when there is a significant imbalance in the distribution of object occurrences among classes. Assigning varying weights to distinct samples according to their distribution throughout the dataset aids in mitigating the problem of foreground-background class imbalance.

The graphs from the YOLOv8m model output illustrate the training process across 50 epochs, focusing on three key loss metrics: box loss, classification loss, and a third metric labelled as dfl loss. The box loss graph shows a significant reduction from an initial value 0.9 to 0.33492, indicating the model's rapidly improving accuracy in bounding box predictions. The classification loss graph starts at a higher initial value of around 3.0 but exhibits a sharp decline to below 0.23533, suggesting the model's increasing proficiency in correctly classifying objects within the bounding boxes. The dfl loss graph presents an initial spike before descending from 1.1 to just above 0.82747, which could reflect the model's capability to adapt to deformations or other specific challenges in object detection Figure 7. These trends are indicative of the model's learning effectiveness and its potential for reliable object detection in various applications. For your research paper, these observations can be highlighted to demonstrate the model's performance and areas for potential improvement.

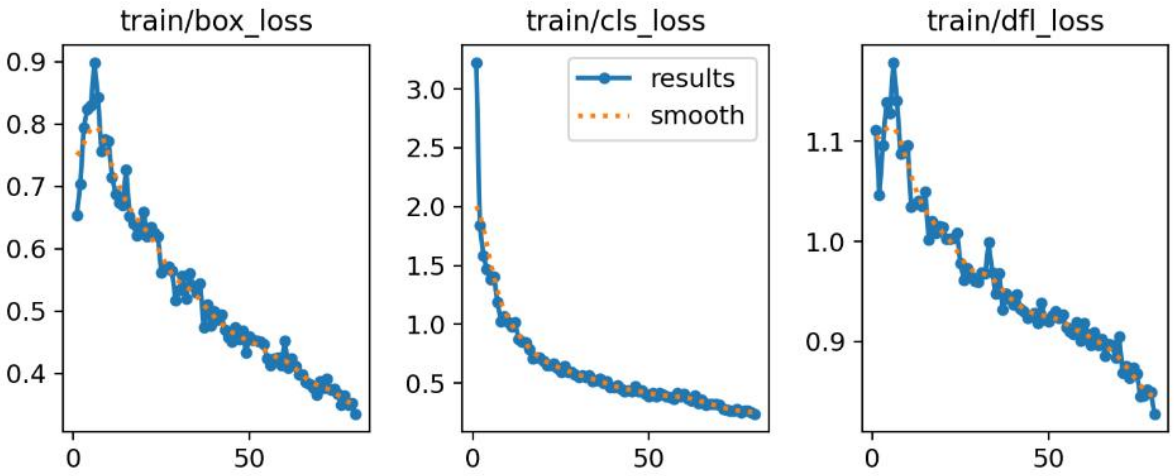

Figure 7: Training loss (box, cls and dfl)

The box loss, graph shows a steep decline in loss, indicating rapid improvement in the model's ability to predict bounding boxes accurately. The classification loss graph demonstrates a dramatic decrease from a high initial value, reflecting the model's enhanced capability in classifying objects within those boxes. The distribution focal loss graph initially spikes, suggesting a potential issue that is quickly resolved as the loss decreases sharply, indicative of the model's learning and adaptation in handling deformations or other complex features Figure 8. These trends are crucial for understanding the model's efficiency and areas for further optimization, providing valuable insights for your research paper on the YOLOv8m model's object detection performance.

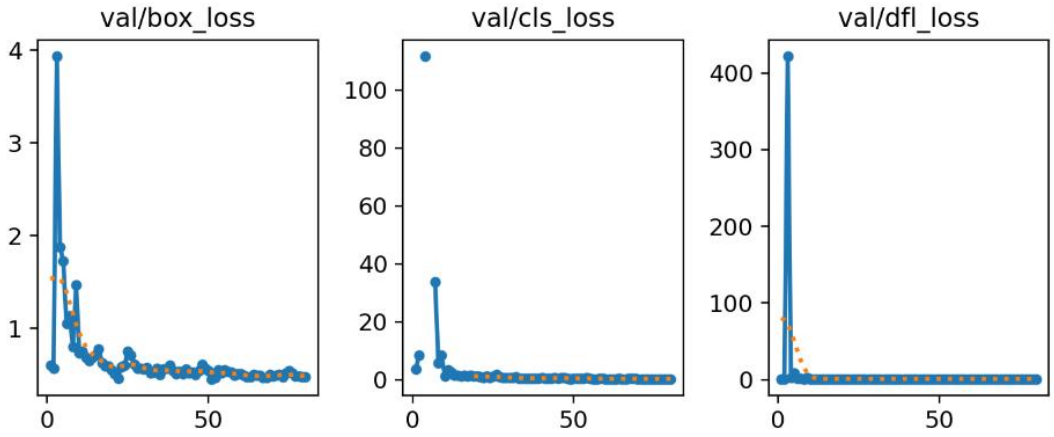

Figure 8: Training loss (box, cls and dfl)

The fundamental indicators of the model's accuracy is Precision, measures the proportion of true positives among all positive predictions, while recall quantifies the ability to detect all actual positives. The precision graph shows that the model maintains high precision, predominantly above 0.8, for a significant range of thresholds before experiencing a decline. This suggests that the model is highly accurate in its predictions, with a low rate of false positives. The recall graph indicates an initial rapid increase to high recall values, stabilizing near 1.0 Figure 9, which implies that the model is capable of detecting nearly all actual positives. These metrics collectively demonstrate the YOLOv8m model's robust performance in identifying objects with high confidence and low false positive rates, making it suitable for real-time applications where accuracy is critical

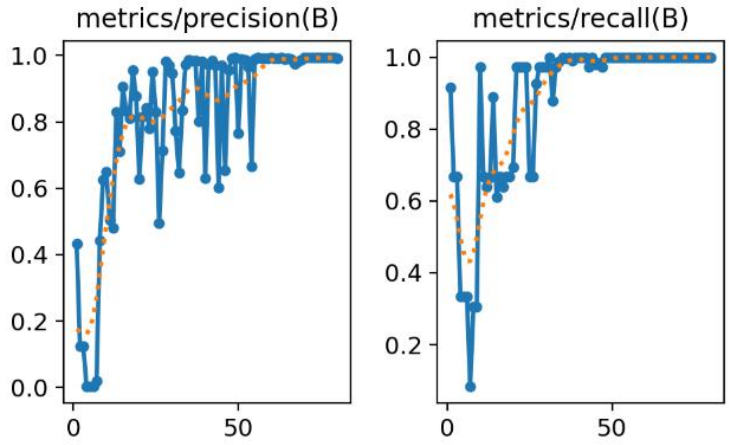

Figure 9: Precision and recall

The x-axis represents the model's confidence level, which is a measure of how certain the model is about its predictions. Higher confidence values indicate that the model is more confident in its classification. Figure 10, the y-axis represents the F1-score, which is a harmonic mean of the model's precision and recall. Precision refers to the proportion of correctly classified detections among all the detections the model makes. Recall refers to the proportion of all actual positive objects that are correctly detected by the model. The F1-score offers a balanced view of how well the model performs, considering both precision and recall. There appears to be a curve for each speed limit (30 km/h, 60 km/h, and 90 km/h), possibly indicating the F1-score of the model for detecting objects at those specific speed limits across different confidence levels. The text "all classes 1.00 at 0.796" might suggest that the model achieves a perfect F1-score (1.00) for all speed limits at a specific confidence level (0.796).

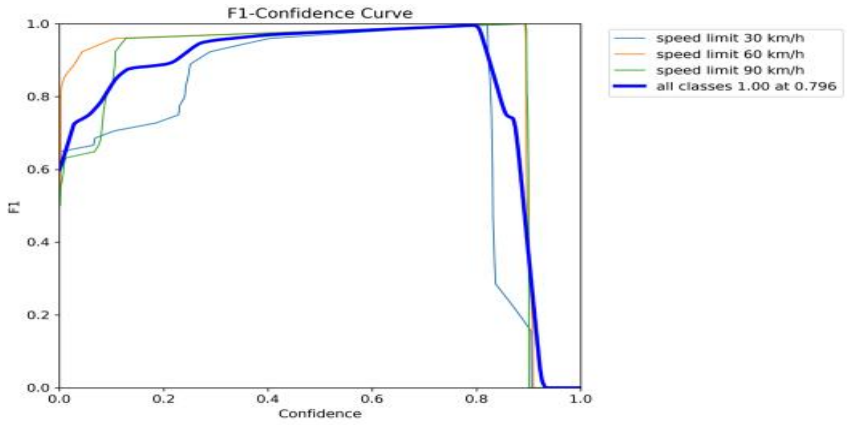

Figure 10: F1-Confidence curve

5.1 Average losses

In object detection models, box loss measures the error in bounding box localization, classification loss quantifies the error in object classification, and dfl loss (likely Distribution Focal Loss) addresses class imbalance by adjusting the loss based on class frequency and detection difficulty.

5.1.1 Validation Loss

The table presents the average loss across three different metrics box loss, classification loss, and dfl loss over the course of 80 epochs. A clear downward trend is observed in all three loss Table. 1 metrics as the epochs progress, indicating effective learning by the model. Specifically, the cls loss shows a significant reduction from 17.2927 in the first epoch to 0.3212 by the eighth epoch, suggesting a substantial improvement in the model's classification accuracy. Similarly, the box loss and distribution focal loss metrics also exhibit a consistent decrease, with the former stabilizing around 0.5 and the latter approaching 0.9. Figure 11, this trend demonstrates the model's increasing proficiency in bounding box prediction and deformation localization, respectively. Overall, the data reflects the model's capability to optimize its parameters

effectively over time, leading to a consistent improvement in performance across all evaluated aspects.

epoch	Val box loss	Val cls loss	Val dfl loss
1	1.3884	17.2927	44.0714
2	0.6627	1.7282	0.9536
3	0.5916	0.9779	0.9191
4	0.5472	0.6247	0.9036
5	0.542	0.5708	0.9
6	0.5108	0.4035	0.898
7	0.4849	0.3547	0.8873
8	0.4967	0.3212	0.8885

Table 1: Validation Loss

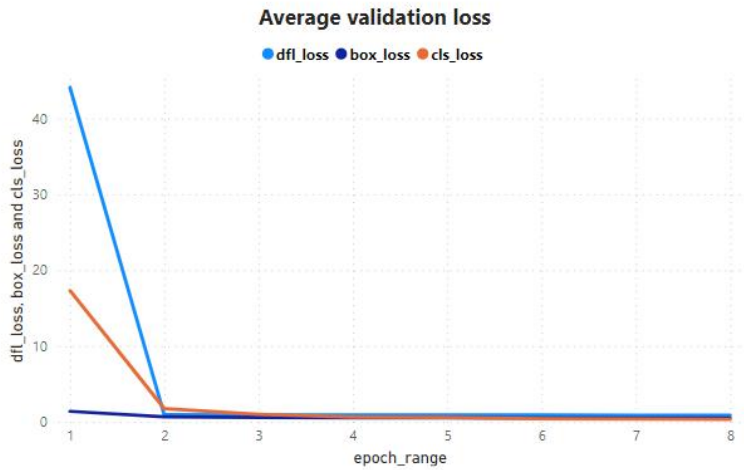

Figure 11: Average of 80 epochs (Validation loss)

5.1.2 Training loss

The data indicates a consistent decrease in loss values across all metrics as the number of epochs increases. This suggests that the model is effectively learning and improving its predictive accuracy over time. Notably, the cls loss, which started at (1.5206) in the first epoch, has seen a substantial reduction to (0.2622) Table. 2 by the eighth epoch, reflecting a significant enhancement in the model's ability to classify correctly. Similarly, the box loss and dfl loss have also shown steady declines, with the former reaching (0.3638) and the latter (0.857) Figure 12, indicating improvements in box localization and deformation prediction, respectively. These trends demonstrate the model's capacity to refine its parameters for better performance as training progresses.

epoch	Train box loss	Train cls loss	Train dfl loss
1	0.7851	1.5206	1.1113
2	0.6681	0.8194	1.0257
3	0.5819	0.6166	0.9815
4	0.5212	0.5206	0.958
5	0.465	0.4374	0.9288

6	0.4359	0.396	0.9176
7	0.3944	0.3398	0.8981
8	0.3638	0.2622	0.857

Table 2: Training Loss

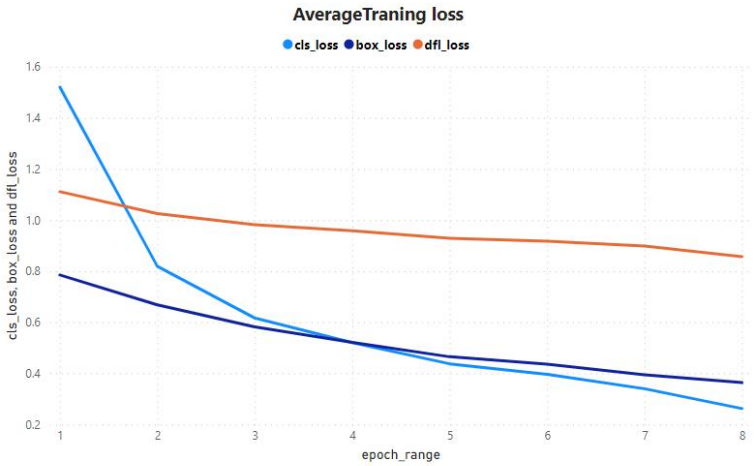

Figure 12: Average of 80 epochs (Training loss)

5.1.3 Precision and recall

Precision and recall are inversely related; as precision increases, recall tends to decrease and vice versa. Precision measures the accuracy of positive predictions, while recall assesses the model's ability to identify all relevant instances. Balancing these metrics is crucial for a model's performance, especially in scenarios where both false positives and false negatives carry significant consequences.

The model's performance has shown a significant improvement over the epochs in terms of both precision and recall. Initially, the precision started at 0.2423 and recall at 0.4917, Table. 3 indicating a moderate detection capability with less than half of the relevant instances being retrieved. However, there was a substantial increase by the second epoch, with precision rising to 0.7532 and recall to 0.6804, suggesting that the model quickly improved its ability to identify relevant instances more accurately Figure 13.

As the epochs progressed, both precision and recall values approached 1, with precision reaching 0.9934 and recall achieving a perfect score of 1 by the eighth epoch. This indicates that the model not only identified almost all relevant instances but also did so with very few false positives. The consistent increase in precision and recall suggests that the model's learning algorithm effectively adapted and optimized its parameters for the task at hand. In conclusion, the data reflects a highly effective model that has achieved near-perfect precision and recall, demonstrating its robustness and reliability in identifying relevant instances with high accuracy by the eighth epoch.

epoch	**Precision**	**recall**
1	0.2423	0.4917
2	0.7532	0.6804
3	0.8323	0.9065
4	0.8592	0.9823

5	0.8859	0.9907
6	0.9559	1
7	0.9884	1
8	0.9934	1

Table 3: Measures of model

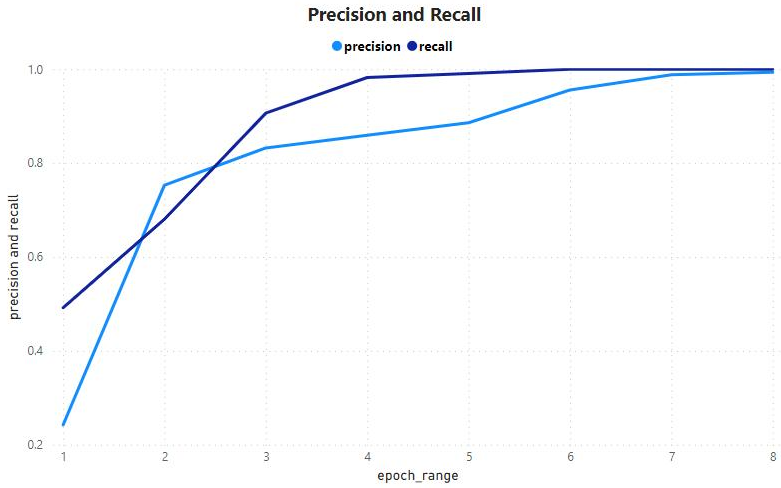

Figure 13: Average of 80 epochs (precision and recall)

5.1.4 mAP

The primary difference between mAP50 and mAP50_95 is the Intersection over Union (IoU) threshold used to evaluate object detection models. mAP50 measures the model's precision at a 50% IoU threshold, indicating a moderate overlap between the predicted and ground-truth bounding boxes. In contrast, mAP50_95 averages the precision across a range of IoU thresholds from 50% to 95%, assessing the model's accuracy at various levels of overlap, which is a more stringent and comprehensive evaluation.

The data indicates a rapid improvement in the model's average precision at detecting objects with a 50% IoU threshold (mAP50) and across a range of thresholds from 50% to 95% (mAP50_95). Starting from an mAP50 of 0.2045 and mAP50_95 of 0.1803 in the first epoch, there is a notable increase by the second epoch to an mAP50 of 0.6861 and mAP50_95 of 0.5882, Table. 4suggesting significant enhancements in the model's detection capabilities Figure 14.

As the training progresses, both metrics show a trend towards stabilization, with the mAP50 reaching a plateau at 0.995 by the seventh epoch, and the mAP50_95 peaking at 0.9047. Interestingly, the mAP50_95 shows a slight decrease in the eighth epoch to 0.8966, which could indicate variations in performance at higher IoU thresholds or potential overfitting to the training data.

In summary, the model demonstrates excellent object detection precision at the 50% IoU threshold and maintains high precision across a spectrum of thresholds. The slight dip in mAP50_95 at the eighth epoch warrants further investigation to ensure the model's generalizability and robustness.

epoch	**mAP50**	**mAP50_95**
1	0.2045	0.1803
2	0.6861	0.5882
3	0.8739	0.7711
4	0.9581	0.8463
5	0.9621	0.8613

6	0.9888	0.8911
7	0.995	0.9047
8	0.995	0.8966

Table 4: Mean Average Precision

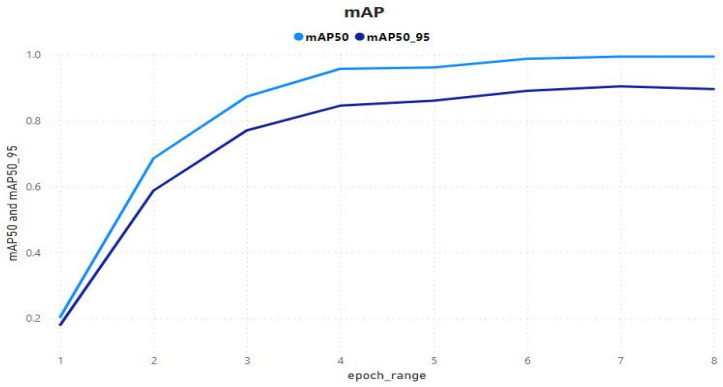

Figure 14: Average of 80 epochs (mAP)

5.1.5 Web cam results

Integrating the YOLO model with a laptop webcam for real-time detection of speed signs involves configuring the webcam to capture video frames at a resolution that the YOLO model can process effectively. As the webcam feeds live video into the system, the YOLO model continuously analyzes each frame to detect and classify speed signs of 30, 60, and 90 km/h. This classification is based on the extensive training the model has undergone, ensuring high accuracy and confidence in identifying the correct speed sign categories. The results are then displayed directly on the live feed, providing immediate visual feedback, as shows in Figure 15. This setup not only allows for real-time analysis, crucial for applications such as driver assistance systems, but also offers the advantages of portability and cost-effectiveness, given that it utilizes the existing hardware of a laptop webcam. The practical application of this integration demonstrates the YOLO model's capability to function effectively in dynamic, real-world scenarios, highlighting its potential in enhancing road safety and traffic management.

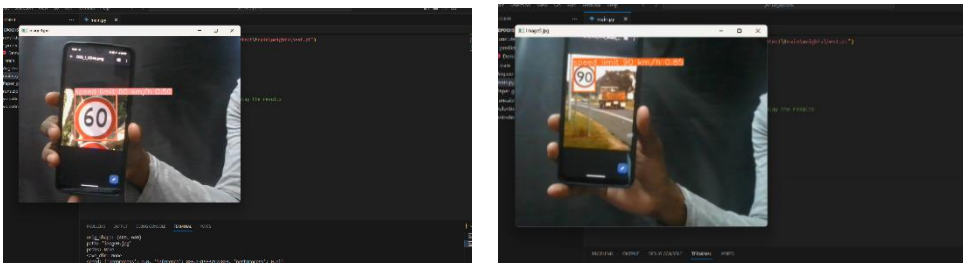

Figure 15: Web cam results of speed limit 60 and 90 km/h

5.1.6 Simulation results

CARLA is an open-source simulator specifically designed for developing, training, and validating autonomous driving systems. Built on the Unreal Engine 4, CARLA offers a realistic virtual environment with customizable urban layouts, vehicles, and weather conditions. Researchers can leverage its open-source code and assets to create tailored scenarios for testing self-driving algorithms Figure 16. CARLA also provides a suite of sensors that mimic real-world data acquisition, enabling the training of perception and decision-making models for autonomous vehicles.

Figure 16: Various speed limit boards in different scenario in CARLA

6. CONCLUSION

In this research, a YOLOv8m model was developed for detecting speed limit signs in autonomous vehicles, using a dataset combining real-world and simulated images enhanced with image augmentation. The model, trained over 80 epochs, demonstrated effective learning and generalizability, indicated by converging loss functions and high precision-recall metrics. Future work includes expanding the dataset to include a wider variety of traffic signs and further testing in simulated environments to ensure robustness and reliability. This study lays the groundwork for implementing a comprehensive speed limit sign detection system in autonomous vehicles, aiming to improve safety and compliance with traffic regulations.

REFERENCES

[1] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings and A. Mouzakitis, "Deep learning-based vehicle behavior prediction for autonomous driving applications: A review", IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 1, pp. 33-47, 2020.

[2] Gibbs G, Jia H, Madani I, "Obstacle Detection with Ultrasonic Sensors and Signal Analysis Metrics", Transp Res Procedia., 28 (2017), pp. 173-182

[3] S. Khalid, J. H. Shah, M. Sharif, M. Rafiq and G. S. Choi, "Traffic sign detection with low complexity for intelligent vehicles based on hybrid features, Computers, Materials & Continua 2023, 76(1), 861-879. https://doi.org/10.32604/cmc.2023.035595

[4] A. Mogelmose, M. M. Trivedi and T. B. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1484-1497, Dec. 2012, doi: 10.1109/TITS.2012.2209421.

[5] L. Aziz, M. S. B. Haji Salam, U. U. Sheikh and S. Ayub, "Exploring Deep Learning-Based Architecture, Strategies, Applications and Current Trends in Generic Object Detection: A Comprehensive Review," in IEEE Access, vol. 8, pp. 170461-170495, 2020, doi: 10.1109/ACCESS.2020.3021508.

[6] M. Rohith, A. Sunil and Mohana, "Comparative Analysis of Edge Computing and Edge Devices: Key Technology in IoT and Computer Vision Applications," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 722-727, doi: 10.1109/RTEICT52294.2021.9573996.

[7] N. Jain, S. Yerragolla, T. Guha and Mohana, "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," 2019 Third International

conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 690-696, doi: 10.1109/I-SMAC47947.2019.9032502.

[8] A. Sunil, M. H. Sheth, S. E and Mohana, "Usual and Unusual Human Activity Recognition in Video using Deep Learning and Artificial Intelligence for Security Applications," 2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2021, pp. 1-6, doi: 10.1109/ICECCT52121.2021.9616791.

[9] Krizhevsky, Alex et al. “ImageNet classification with deep convolutional neural networks.” Communications of the ACM 60 (2012): 84 - 90.

[10] V. Rajesh, U. P. Naik and Mohana, "Quantum Convolutional Neural Networks (QCNN) Using Deep Learning for Computer Vision Applications," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 728-734, doi: 10.1109/RTEICT52294.2021.9574030.

[11] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., et al. (2016). End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316.

[12] A. Agafonov and A. Yumaganov, "3D Objects Detection in an Autonomous Car Driving Problem," 2020 International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russia, 2020, pp. 1-5, doi: 10.1109/ITNT49337.2020.9253253.

[13] D. R. Niranjan, B. C. VinayKarthik and Mohana, "Deep Learning based Object Detection Model for Autonomous Driving Research using CARLA Simulator," *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2021, pp. 1251-1258, doi: 10.1109/ICOSEC51865.2021.9591747.

[14] N. Rao and S. Sundaram, "Spatio-Temporal Look-Ahead Trajectory Prediction using Memory Neural Network," *2021 International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9534209.

[15] Shen, Y.; Wijayaratne, N.; Du, P.; Jiang, S.; Driggs-Campbell, K. AutoPreview: A Framework for Autopilot Behavior Understanding. In Proceedings of the Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021; pp. 1–6.

[16] A. Mangawati, Mohana, M. Leesan and H. V. R. Aradhya, "Object Tracking Algorithms for Video Surveillance Applications," *2018 International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2018, pp. 0667-0671, doi: 10.1109/ICCSP.2018.8524260.

[17] Mohana and HV Ravish Aradhya, “Object Detection and Tracking using Deep Learning and Artificial Intelligence for Video Surveillance Applications” International Journal of Advanced Computer Science and Applications(IJACSA), 10(12), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0101269

[18] H. V. Ravish Aradhya, Mohana and Kiran Anil Chikodi, "Notice of Removal: Real time objects detection and positioning in multiple regions using single fixed camera view for video surveillance applications," *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, Visakhapatnam, India, 2015, pp. 1-6, doi: 10.1109/EESCO.2015.7253654

[19] Megalingam, R. K., Thanigundala, K., Musani, S. R., Nidamanuru, H., & Gadde, L. (2023). Indian traffic sign detection and recognition using deep learning. International Journal of Transportation Science and Technology, 12(3), 683-699. https://doi.org/10.1016/j.ijtst.2022.06.002

[20] A. Agnihotri, P. Saraf and K. R. Bapnad, "A Convolutional Neural Network Approach Towards Self-Driving Cars," *2019 IEEE 16th India Council International Conference (INDICON)*, Rajkot, India, 2019, pp. 1-4, doi: 10.1109/INDICON47234.2019.9030307

[21] M. D. RADU, I. M. COSTEA and V. A. STAN, "Automatic Traffic Sign Recognition Artificial Inteligence - Deep Learning Algorithm," *2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, Bucharest, Romania, 2020, pp. 1-4, doi: 10.1109/ECAI50035.2020.9223186.

[22] Y. Sun, P. Ge and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," *2019 Chinese Automation Congress (CAC)*, Hangzhou, China, 2019, pp. 2851-2854, doi: 10.1109/CAC48633.2019.8997240.

[23] Handoko, J. H. Pratama, and B. W. Yohanes, “Traffic sign detection optimization using color and shape segmentation as preprocessing system,” Telkomnika (Telecommunication Computing Electronics and Control), vol. 19, no. 1, pp. 173–181, 2021, doi: 10.12928/TELKOMNIKA.V19I1.16281.

[24] M. Çetinkaya and T. Acarman, "Traffic Sign Detection by Image Preprocessing and Deep Learning," *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2021, pp. 1165-1170, doi: 10.1109/ICICCS51141.2021.9432088.

[25] H. Gholamalinejad and H. Khosravi, "Irvd: A large-scale dataset for classification of iranian vehicles in urban streets", *Journal of AI and Data Mining*, vol. 9, no. 1, pp. 1-9, 2021, [online] Available: https://jad.shahroodut.ac.ir/article1789.html.

[26] U. P. Naik, V. Rajesh, R. K. R and Mohana, "Implementation of YOLOv4 Algorithm for Multiple Object Detection in Image and Video Dataset using Deep Learning and Artificial Intelligence for Urban Traffic Video Surveillance Application," *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Erode, India, 2021, pp. 1-6, doi: 10.1109/ICECCT52121.2021.9616625.