AI-driven Autonomous Vehicle using Yolov8 and Deep learning

MSc Research Project
MSc in Artificial Intelligence

Kshitija Chavan
Student ID: x22123598

School of Computing
National College of Ireland

Supervisor: Abdul Razzaq

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Kshitija Chavan

Student ID: x22123598

Programme: MSc in Artificial Intelligence **Year:** 2023

Module: Research Project

Supervisor: Abdul Razzaq

Submission Due Date: 31st January, 2024

Project Title: AI-driven Autonomous Vehicle using Yolov8 and Deep learning

Word Count: 6000 **Page Count:** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Kshitija Chavan

Date: 31st January, 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

AI-driven autonomous Vehicle using Yolov8 and Deep learning

Kshitija Chavan

x22123598

Abstract

By automating vehicle tracking and control, advanced driver assistance systems, or ADAS, have emerged as an essential component of technology for road safety. The inherent difficulties in developing ADAS are addressed in this study, which offers a thorough remedy that combines deep learning and computer vision techniques. With the help of real-time YOLOv8 object detection, OpenCV-based lane detection, and depth-based distance estimate, the system guarantees safe vehicle following, automated steering assistance, and accurate identification of cars, pedestrians, and barriers. Using eye blink and yawning tracking, a convolutional neural network detects driver fatigue and triggers appropriate alerts. Extensive testing in diverse traffic settings has confirmed the system's effectiveness in detecting drowsiness and preventing collisions.

The suggested ADAS not only reduces accident risks but also marks a substantial development towards accessible, sensor-independent vision-based ADAS solutions, advancing the field of road safety technologies by seamlessly fusing computer vision-based automation with driver monitoring.

Keywords: Advanced Driver Assistance Systems (ADAS), Computer Vision, YOLOv8, Deep Learning, Object Detection, OpenCV, Lane Detection, Distance Estimation, Convolutional Neural Network (CNN), Automated Steering Assistance, Drowsiness Monitoring.

1. Introduction

The domain of traffic safety is progressively reliant on Advanced Driver Assistance Systems (ADAS), which mechanize vital operations including vehicle surveillance and management. But the creation of strong ADAS faces difficulties with perception and control, therefore creative solutions are needed to guarantee their efficacy. Although ADAS has advanced, there are still gaps in the field when it comes to effectively combining computer vision and deep learning techniques to handle challenging road circumstances.

Improving ADAS effectiveness can reduce risks and avoid accidents through automation, as there are more than 1.3 million fatalities worldwide each year. Real-time vision-based ADAS advancements meet accessibility and safety requirements that are not satisfied by current sensor-based systems that demand specific hardware.

The research question posed in this study investigates how the combination of computer vision can and learning techniques enhance the effectiveness of Advanced Driver Assistance Systems (ADAS) in promoting road safety.

The following sets of study objectives were developed to address this question:

1. Examine the current state of the art in general about the ADAS integration of deep learning and computer vision.

2. Create a complete ADAS system with depth-based distance estimation, OpenCV-based lane detection, and real-time YOLOv8 object detection.
3. Put the intended ADAS system into practice and use GPUs to optimize its real-time performance.
4. Examine how well the integrated ADAS performs in a variety of driving circumstances, paying particular attention to collision prevention and prompt driver fatigue detection.

The main contribution is providing automated control and driver monitoring based solely on computer vision, which replaces the need for specialized sensors through technical optimization and delivers vital ADAS capability.

2. Related Work

In the following years, completely autonomous vehicles are expected to play a major part in the advancement of transportation systems. Self-driving cars have several notable advantages, but two that stand out are enhanced safety and optimized traffic control. These driverless vehicles offer safer and more convenient travel experiences, giving people who were previously limited by age or handicap more freedom. With its revolutionary potential, autonomous driving technology has the potential to completely change accessibility and mobility on public roadways.

2.1 Vehicle Detection

Scholars have investigated a variety of tactics in an effort to improve vehicle detection in bad weather. To lessen the effects of bad weather, vision-based technologies use complex image preparation techniques including neural networks, median filtering, and histogram equalization. The goal of multifeature fusion techniques is to provide a more comprehensive impression of vehicles by addressing the issues caused by the disappearance of single-vehicle features. Furthermore, enhanced deep learning models that incorporate image augmentation and denoising techniques are created with weather adaptability in mind. In addition to vision-based techniques, researchers are working on denoising, data augmentation, and model enhancements to get over unfavorable weather constraints. Radar and lidar provide strong alternatives. In order to provide reliable identification across a variety of driving conditions, (Wang, Wang, & Gao, 2021) contend that intelligent vehicle detection has evolved by merging several sensors, including radar, lidar, cameras, and infrared images through integration schemes fusing the individual strengths.

Edge intelligence and artificial intelligence integration has gained popularity recently, especially in the context of intelligent transportation systems. In order to improve the processing speed of video surveillance systems in ITS utilizing an improved-YOLOv4 vehicle detection algorithm, Chen et al. (2023) emphasize the deployment of EI at the network edge. This technique significantly improves accuracy by combining a high-resolution network (HRNet) for increased vehicle detection with an efficient channel attention (ECA) mechanism. In the meantime, Xu, Wang, and Gu (2019) make a contribution by presenting the VEDAI dataset, an important tool for identifying aerial vehicles in satellite photos. The dataset, which consists of nine vehicle classifications, is used as a standard to assess detection methods.

Furthermore, Kosuru and Venkitaraman (2022) tackle the problems associated with picture filters in autonomous cars by utilizing YOLOv2 algorithm techniques and putting forth a low-cost noise cancellation method. Their work focuses on enhancing picture clarity through the use of Median-Filtration and pre-learning models to improve accuracy. They also anticipate and address potential errors in autonomous driving systems, particularly in challenging environments. Together, these studies show the progress and difficulties in vehicle detection, covering everything from algorithmic improvements to the development of benchmark datasets and solving real-world application issues.

2.2 Lane Detection

Researchers have made great progress in improving the precision and effectiveness of detection algorithms in recent lane detecting literature. Using a variety of video action recognition techniques, (Biparva et al., 2022) tackled lane-change classification and target vehicle prediction, producing noteworthy results with varying ROI sizes and observation horizons. In order to forecast lane changes early on, (Mozaffari et al., 2022) developed a multi-task attention-based CNN model that outperformed previous techniques and showed enhanced short- and long-term prediction performances. In their systematic literature analysis from (Zakaria et al., 2023) highlighted the variety of approaches used for lane detection and demonstrated how deep learning techniques like CNNs, FCNs, and RNNs were integrated. Deep learning combined with attention processes was found to be a popular approach that showed promise for enhancing detection performance overall. With a focus on pedestrian lane detection, (Lei et al., 2022) provided a thorough analysis of both conventional and deep learning techniques, such as border- and color-based methods. By addressing issues in practical situations and creating new research opportunities, these works jointly advance lane detection technologies and enhance the effectiveness and performance of lane detecting systems.

2.3 Drowsiness Detection

The literature on driver sleepiness detection systems highlights the crucial role that technology plays in improving road safety by providing a thorough overview of a variety of approaches. An extensive analysis of current systems was carried out by (Ojha et al., 2023), who indicated the lack of a universal solution by pointing out the many advantages and disadvantages of each strategy. They acknowledge the potential of a CNN-based sleepiness detection system to lower fatigue-related accidents. A comparative analysis by (Khadkikar et al., 2023) highlights the extensive range of features investigated in addressing driver weariness, with CNN, SVM, and aspect ratio models standing out. (T.S et al., 2023) present a Drowsiness Alert System and Remote Healthcare Monitoring, highlighting its innovative approach to enhancing both road safety and healthcare results.
The prediction approaches are thoroughly examined by (P.S. et al.,2022), who also note the ad vantages and disadvantages of each. In their exploration of the difficulties associated with drowsiness research, (Perkins et al., 2023) support the use of prediction models, multi-modal data, and address problems with data quality. The body of research highlights the continuous attempts to improve and develop driver sleepiness detection systems for practical use, with the goal of enhancing overall driver safety and saving lives.

2.4 Steering Wheel Automation

The evaluated literature covers a wide range of topics related to driver assistance and autonomous navigation, providing insight into important approaches and advancements in technology. The wide variety of utilizes that self-driving robots have in the industry was emphasized by (Singh et al., 2022) as they investigated the several methods that these machines use for wheel synchronization, evading obstacles, and trajectory following. During extended, monotonous driving duties, (Wang et al., 2017) concentrated on According to the study, driving performance improved when activated haptic assistance was used, suggesting that this technology has the potential to be used as a driving assistant to reduce passive tiredness. The importance of sensor and sensor fusion technologies in allowing autonomous cars to sense and comprehend their surroundings was emphasized by (Kumar et al., 2023).

2.5 Advanced Driver Assistance System

The topic of Advanced Driver Assistance Systems (ADAS) has witnessed notable advancements in research over the past ten years, especially with the approaching introduction of autonomous vehicles (Machaiah & Pavithra, 2022). This study covers a broad overview of different lane detection and tracing methods and sheds light on the few studies that have been done on these important topics. Because learning-based approaches are so effective in real-time circumstances, they become essential. While the lack of datasets is still a widespread problem, integrating artificial sensor data is suggested as a workaround for algorithm testing. Future research directions are identified by the review, which highlights the need for studies on weather-resistant reliability, complicated roads, and managing unstructured road settings.

After conducting an initial investigation into the ADAS/AD simulation ecosystem, the study finds 72 simulators that were cited in 13 surveys. The most often used simulators are CARLA, Airsim, and SUMO (Koroglu & Wotawa, 2023). The number of new simulators is growing at a rate that is quadratic, which calls for a systematic review to handle the growing ADAS/AD simulation domain in detail. Subsequent efforts will classify simulators according to their nature, supporting sensors, intended ADAS/AD functionality, and conformity to standards. The paper also emphasizes how many simulators are interrelated, showing examples of how more recent simulators build upon earlier systems. In order to provide a more extensive and in-depth understanding of the ADAS/AD simulation landscape, the assessment also anticipates a thorough investigation into the features and tradeoffs inherent in various simulators.

2.6 Yolov8

The combined literature study offers a thorough synopsis of the various uses and developments associated with the YOLO (You Only Look Once) algorithm. (Uma et al., 2023) carry out a comprehensive analysis of YOLO and augmented reality (AR), highlighting its application for improved object recognition in industries such as gaming, retail, healthcare, education, and the military. The report emphasizes difficulties with augmented reality and object recognition, such as hardware specifications, UI design, privacy issues, and spatial alignment. The study also examines the potential and synergies that arise from merging AR with YOLO, laying the groundwork for further investigation and advancement in this area.

In-depth research on YOLO and augmented reality (AR) is done by (Uma et al., 2023), who highlight how these technologies can be integrated for improved object recognition in industries like gaming, education, healthcare, retail, and the military. The report highlights issues with hardware specifications, UI design, privacy, and spatial alignment that are associated with augmented reality and object recognition. Furthermore, the study investigates the opportunities and synergies presented by fusing AR with YOLO, laying the groundwork for further investigation and advancement at this nexus.

In their thorough analysis of deep learning-based object detection, (Amjoud and Amrouch, 2023) divide models into four primary categories: transformer-based detectors, one-stage anchor-based detectors, anchor-free detectors, and two-stage anchor-based detectors. Through the evaluation of these models on large item detection databases, the study reveals how object detection speed and accuracy are changing. In addition to resolving the speed-accuracy trade-off, the authors highlight prospects in object detection research, including investigating 3D object identification, multi-modal object detection, overcoming microscopic item detection issues, and enhancing few-shot learning. Important new information about action detection pipelines is provided by (Alikhanov and Kim, 2023), especially for surveillance scenarios. The study emphasizes how crucial it is to choose trackers and configurations carefully in order to improve the effectiveness of action detection systems in surveillance applications. The study emphasizes how important it is to assess multi-object trackers in authentic surveillance scenarios and how important it is to choose the right dataset, adjust the detector, and adjust the input resolution in order to get precise tracking outcomes. In order to identify plants in agricultural plantations, (Grujev et al., 2023) concentrate on using object recognition and classification techniques, particularly YOLO, in aerial photos. Reviewing a number of scientific studies that apply algorithms to UAV-captured photos, the paper highlights the advantages of doing so, including the ability to differentiate between cultivated and weed species, evaluate plant health, and determine fruit development. In these applications, deep learning algorithms like as DBN, CNN, YOLO, and Mask R-CNN are frequently highlighted by the authors, who also confirm that high plant detection and classification accuracy is attained.

3. Research Methodology

3.1 Overview of Methodology

To incorporate vehicle recognition, drowsiness detection, and an intuitive graphical user interface, the methodology employs a systematic approach. YOLO, Tkinter, OpenCV, dlib, Python, and a thorough literature review all influenced the technologies chosen. Facial landmarks and an EAR algorithm are used in drowsiness detection to send out email notifications. A real-time bounding box extraction YOLO model is used for vehicle detection. Incorporating alarm triggers depending on vehicle proximity and tiredness guarantees smooth communication. Facilitating user interaction is a Tkinter GUI with safe login. Using metrics for performance like precision and accuracy, real-time video frames are collected for data analysis. The approach prioritizes moral issues and allows for future development flexibility.

3.3 Data Collection

A crucial first step towards facilitating successful deep learning deployments is data preparation. The dataset that was used comprises many picture files located in different folders that capture car steering wheel orientations from various perspectives. Python packages programmatically link pertinent files and annotations to streamline access. There are training and validation sets inside the picture dataset. Each picture is indexed with its associated steering angle values in degrees in an attached text file; they are then converted to radians to aid in optimization during neural network processing. Deeper propagation of image characteristics through successive network layers is made possible at model runtime by activation functions such as ReLU and tanh, which improve pattern recognition and regression capabilities. Overall, systematic metadata collection, reformatting, and structuring along with content feeds improves the trainability of intricate deep neural network designs while reducing the amount of labor-intensive human data wrangling. The focus on preprocessing and indexing makes it easier to feed driving footage into certain machine learning frameworks in a reproducible manner, which is necessary to efficiently derive insights about vehicular control from raw dashcam pictures.

3.2 Data Preprocessing

For the integrated algorithms, a thorough preparation pipeline consists of a number of carefully planned stages, each of which is tailored to the unique needs of the various functions that are contained in the system. For robust face detection in the drowsiness detection module, the code carefully applies Haarcascades, extracts facial landmarks that are important for measuring eye movements using the dlib library, and measures the degree of eye openness using the Eye Aspect Ratio (EAR) metric. This guarantees a thorough comprehension of the subtleties in eye movements and facial expressions, which serves as the basis for efficient drowsiness detection.

Meanwhile, the part that detects vehicles starts when the YOLO model, which is able to identify objects in pictures, is initialized. In order to comply with the model's input size requirements, it applies image resizing techniques. After that, it moves on to vehicle detection, estimating distances given focus length and object height. Contributing to an all-encompassing framework for automobile safety, this distance computation is essential in determining proximity alarms.

The GUI and login modules go through their own preprocessing stages in order to facilitate user interaction. It coordinates the user credential verification process, guaranteeing a safe and verified access environment. This module also sets up button functions, associates actions with button clicks, and creates a graphical user interface (GUI) that is both aesthetically pleasing and easy to use by adjusting button and picture display settings.

3.3 Algorithm/Model Selection

Vehicle Detection: As Yolov8 model can detect objects in real time, it is used for vehicle detection. Because of its speed and accuracy, Yolov8 model is a good fit for processing video streams, guaranteeing accurate and timely vehicle identification in a variety of settings.

Drowsiness Detection: For drowsiness detection, the method combines the eye aspect ratio (EAR) computation with facial landmark detection via the dlib package. This decision is supported by the ease and efficacy of EAR as a trustworthy metric for sleepiness assessment, as well as the effectiveness of facial landmarks in collecting important aspects associated to eye movements.

Lane Detection: Utilizing techniques from computer vision, such as Hough transform, region of interest masking, and edge detection, the lane detection module functions. Given these systems' accuracy in identifying lane markers in various traffic circumstances, the choice is reasonable.

Graphical User Interface (GUI) and User Authentication: The Tkinter library is the main source of dependency for GUI development and simple user interfaces in the user authentication and GUI modules. Tkinter's simplicity and ease of interaction with the Python programming environment, which make it appropriate for developing an interface that is easy to use, serve as justification for this decision.

3.4 System Configuration and Setup

The system is designed to facilitate a comprehensive solution for real-time vehicle monitoring and alerts. The system is started via the user interface (UI), which is constructed using the Tkinter framework. Before granting access to the primary functions, an authentication module improves security. For real-time video collection and processing, the video processing component use OpenCV.

The system incorporates the environment perception modules, which include lane detection, vehicle detection, and distance estimate. These modules assess the road surroundings, detect lane lines, recognize vehicles, and estimate distances using computer vision techniques.

A vital safety component, drowsiness detection, analyzes the driver's eye behavior using facial landmarks and alarms if indications of tiredness are detected. Pygame and PyAudio are used in the alert and steering stimulation modules to send auditory alerts and stimulate the steering mechanism, assuring fast driver response.

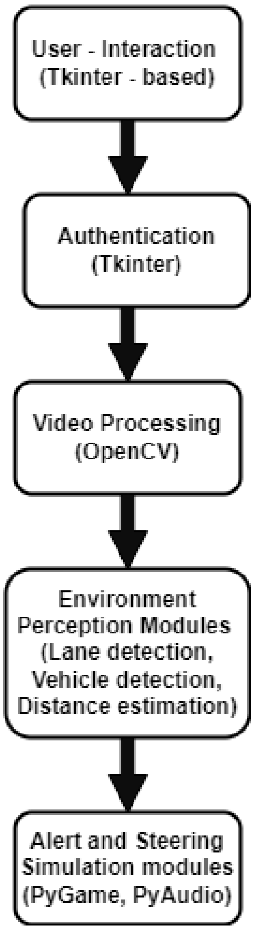


Figure 1. Flowchart

4. Design Specification

User Interface and Authentication: To enable user registration and validation for restricted system access, a frontend interface is created in the Tkinter Python Module. Core dashboard panels offer elements for managing video streaming after login. Runtime metadata logging can be tracked with the additional analytics view.

Video handling and preprocessing: After being converted to appropriate frames, input movies including driver and road footage are set up to feed into OpenCV video capture pipelines, which are then fed into computer vision modules. Stabilization, noise reduction, and related improvements are applied to images to improve processing afterward.

Lane and Vehicle Detection: Using a deep neural network powered by TensorFlow backend, the transformed imagery is evaluated to identify lane markings and detect vehicles and pedestrians with high accuracy. Strong perception is essential for making decisions about navigation and control.

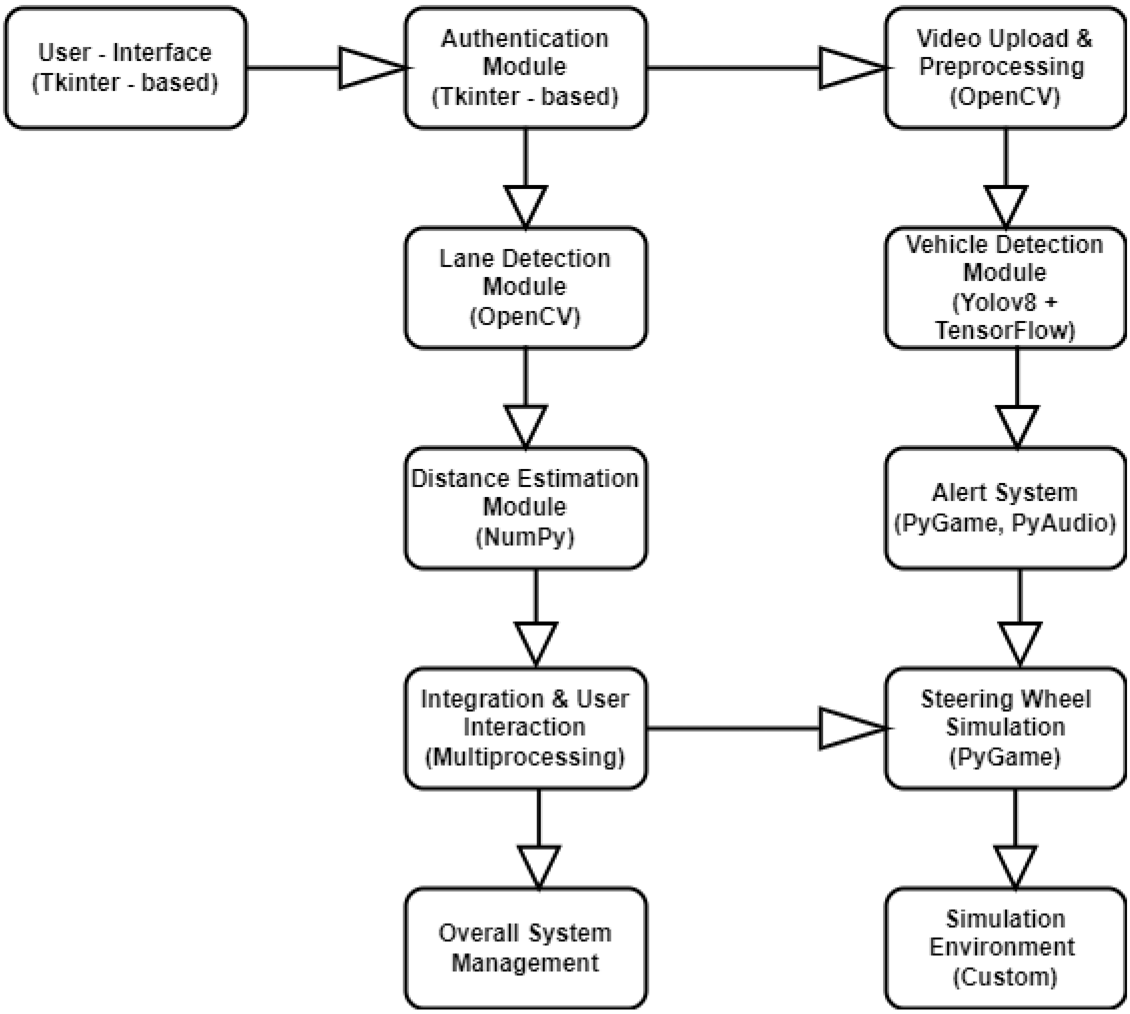


Figure 2. Architecture

Distance Estimation: By using a single monocular camera, NumPy's synthetic depth map generating capabilities can provide accurate estimates of the depth and scale of nearby vehicles. This makes it easier to avoid tailgating by controlling brakes.

Notifications and Alerts: The PyGame and PyAudio Python modules provide for the real-time notification of users through logging and the activation of visual and audio alerts, respectively, based on detected distractions.

Control, Actuation, and Visualization: PyGame makes it easier to relay control signals to a virtual steering wheel with force-feedback capability, simulating real-world driving circumstances. This is done through the use of GPIO-based electronics. It facilitates visualizing.

System Management and Integration: To fulfill real-time performance restrictions, all of the modules for video handling, environment perception, distance mapping, alarm modulation, and steering control interface are smoothly integrated in a parallelized fashion using Python's multiprocessing structures. An orchestration layer based on Python scripts oversees sequencing and vehicle integration, enabling programmatic coordination and data transmission between components.

5. Implementation

The work is implemented by integrating multiple modules: OpenCV is employed for video processing, YOLOv8 and TensorFlow are utilized for object detection, Dlib is used for drowsiness detection, and Tkinter is used for user interface and authentication. Pygame and PyAudio is used by the system for alert production and interaction. The primary objective of the work is to integrate components of a cohesive system to improve road safety by detecting fatigue and providing alarm mechanisms.

5.1 Programming Language

Python is used to implement the code because of its versatility and large packages for Machine Learning (ML) and computer vision.

5.2 Libraries and Frameworks

OpenCV: This computer vision library offers a wide range of features for graphical overlays, feature detection, streaming video, image modification, and other frame processing applications. Alongside mission-critical vision tasks like applying lane detection algorithms and identifying facial landmark coordinates connected to fatigue metrics, we use OpenCV for generics operations like input feeding and formatting.

TensorFlow: This all-in-one machine learning platform allows for the training and deployment of deep neural networks. We build, optimize, and run-time execute the YOLOv8 architecture using TensorFlow to provide reliable real-time automotive entity recognition and localization, including cars, buses, pedestrians, and other vehicles.

Dlib: This toolkit makes it possible to put cutting-edge machine learning and computer vision algorithms into practice. It is especially helpful for recognizing and tracking features of the human face. We enable tracking eye blink durations and yawning patterns indicating to reduced driver awareness by utilizing Dlib for facial landmark estimation.

PyGame: This graphics toolkit facilitates the creation of interactive animations and user interfaces and is well-suited for creating multimedia applications. We utilize PyGame to provide visual warnings that remind users to pay attention and to replicate steering wheel reactions, enabling hands-free vehicle operation.

PyAudio: This Python package facilitates the extraction of audio streams and the creation of alert sound cues. When sleepiness episodes are recognized, we use PyAudio techniques to enable audio warnings urging drivers to overcome them.

Tkinter: This standard GUI package provides back-end tools for event sequencing and front-end program creation. In addition to providing system authentication, Tkinter makes it simple to integrate all peripherals into a single dashboard that features a live video feed and log/analytics displays.

5.3 Model Implemention

5.3.1 Facial Landmark Detection:

The dlib package, a potent toolset for machine learning and computer vision applications, is used in the facial landmark identification procedure. Facial landmarks, or key spots on the face like the outer edges of the eyes and the tip of the nose, are detected using the dlib library. The eye aspect ratio (EAR), a key parameter in the identification of tiredness, must be calculated using these landmarks.

5.3.2 Vehicle Detection:

Modern real-time object detection technology, the YOLOv8 model, is integrated for vehicle detection. YOLOv8 is driven by the recognized machine learning framework TensorFlow. The model analyzes video frames and uses real-time vehicle identification and location to provide a thorough picture of the road environment.

5.3.3 Lane Detection:

Lane detection uses OpenCV, a flexible computer vision framework. To identify elements associated to lanes, different image processing techniques are applied to video frames. Lane boundary detection algorithms are used to help with activities like lane-keeping assistance and to improve perception of the road structure.

5.3.4 Distance Estimation:

The process of estimating distance involves computing the separations between identified objects, mostly cars. For a precise estimate of distance, the camera's focal length and the height of the object (vehicle) in the image framework are used. When determining how close other items are to you on the road, this knowledge is essential. When determining how close other items are to you on the road, this knowledge is essential.

5.3.5 Drowsiness Detection:

Monitoring the eye aspect ratio (EAR), which is derived from facial landmarks, is necessary for drowsiness detection. The EAR thresholds are designed to identify whether the driver is showing signs of tiredness or has their eyes open or closed. This module makes considerable use of OpenCV for image processing and face landmark extraction.

5.3.6 Alert System and Steering Simulation:

For timely alerts, an integrated alert system is put into place. PyAudio is used to offer auditory alerts, and Pygame is used to display messages on the screen. Furthermore, Pygame is used to integrate a steering wheel simulation, enabling prompt and accurate remedial actions in the event of drowsiness or potential risks**.**

6. Evaluation

The produced system's overall functioning and effectiveness are thoroughly examined during the evaluation phase. The user experience, scenario simulations, and practical evaluations are used to assess the adaptability and performance of the system. The evaluation's goal is to confirm that the system can accurately identify and address possible risks associated with driving while intoxicated. The evaluation's conclusions offer insightful information about the system's practicality and point out possible areas for enhancements or optimizations.

6.1 Evaluation Metrics

- Accuracy: Reflects the general accuracy of the system's predictions, especially when it comes to recognizing tiredness, spotting cars, and precisely measuring distances while processing real-time footage.

- Precision: Assesses the accuracy of positive forecasts, essential for reducing false alarms in the alert system and guaranteeing that alerts are set off by intoxication or possible accident scenarios.

- Recall (Sensitivity): Evaluates how well the system recognizes all signs of fatigue and other threats, making sure that no important circumstances are missed.

- F1 Score: The F1 Score is a complete evaluation of the system's overall effectiveness in terms of alerting to tiredness and probable crashes. It is a balanced data that requires into consideration both precision and recall.

6.2 User Interface

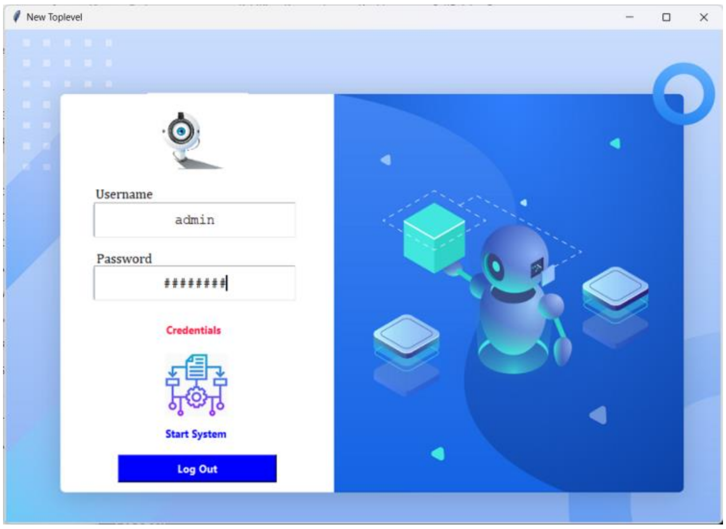


Figure 3. Login Page

The user interface (UI) highlights a particular set of features. Before getting access to the system, users must log in using credentials. A safe and authenticated user experience is guaranteed by this login process. After logging in successfully, users may submit a picture or a video using the user interface. An essential part of the project's functionality is the upload capability, which lets users enter visual data into the system for processing or analysis. By requiring a login, the UI's picture and video uploading features are made accessible only to authorized users, providing an extra degree of protection and control.

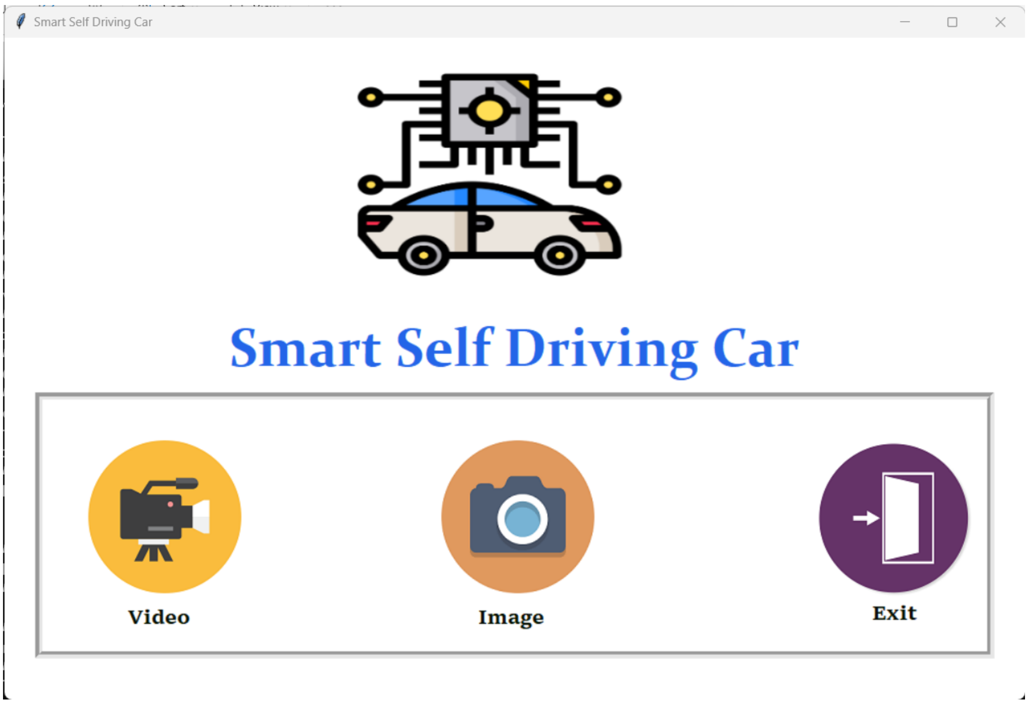


Figure 4. Dashboard

6.3 Results

```
284/284 [==============================] - 1s 3ms/step
Precision: 0.763170188220263
Recall: 0.8735961242017177
F1 Score: 0.8146581628369097
Accuracy: 0.8735961242017177
```

Figure 4. Metrics of the Model

The model's precision score of 0.76 shows how dependable it is in identifying good cases, while its recall score of 0.87 shows how successfully it detects a sizable fraction of actual positive events. With an F1 score of 0.81, the model displays thorough precision and recall. The model's ability to identify objects effectively is illustrated by its notable accuracy of 0.87, which is vital for real-time applications like driver assistance systems. The model's learning trajectory is clearly represented by the accuracy vs. epoch graph (Figure 4) that goes along with it, demonstrating how it continues to improve across training epochs. This highlights the model's flexibility and proficiency in handling the intricacies of the given tasks, guaranteeing strong and dependable performance in real-world situations.

```
Epoch 1/5
9078/9081 [============================>.] - ETA: 0s - loss: 0.2987 - accuracy: 0.8746
Epoch 1: val_loss improved from inf to 0.28479, saving model to Stearing.h5
9081/9081 [==============================] - 86s 8ms/step - loss: 0.2987 - accuracy: 0.8746 - val_loss: 0.2848 - val_accuracy:
0.8736
Epoch 2/5
  10/9081 [..............................] - ETA: 51s - loss: 0.2324 - accuracy: 0.9250
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 fil
e via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.sav
e('my_model.keras')`.
  saving_api.save_model(
9078/9081 [============================>.] - ETA: 0s - loss: 0.2984 - accuracy: 0.8748
Epoch 2: val_loss improved from 0.28479 to 0.28458, saving model to Stearing.h5
9081/9081 [==============================] - 66s 7ms/step - loss: 0.2984 - accuracy: 0.8748 - val_loss: 0.2846 - val_accuracy:
0.8736
Epoch 3/5
9076/9081 [============================>.] - ETA: 0s - loss: 0.2981 - accuracy: 0.8748
Epoch 3: val_loss improved from 0.28458 to 0.28452, saving model to Stearing.h5
9081/9081 [==============================] - 64s 7ms/step - loss: 0.2981 - accuracy: 0.8748 - val_loss: 0.2845 - val_accuracy:
0.8736
Epoch 4/5
9075/9081 [============================>.] - ETA: 0s - loss: 0.2985 - accuracy: 0.8747
Epoch 4: val_loss did not improve from 0.28452
9081/9081 [==============================] - 66s 7ms/step - loss: 0.2983 - accuracy: 0.8748 - val_loss: 0.2849 - val_accuracy:
0.8736
Epoch 5/5
9079/9081 [============================>.] - ETA: 0s - loss: 0.2980 - accuracy: 0.8748
Epoch 5: val_loss improved from 0.28452 to 0.28446, saving model to Stearing.h5
9081/9081 [==============================] - 62s 7ms/step - loss: 0.2981 - accuracy: 0.8748 - val_loss: 0.2845 - val_accuracy:
0.8736
```

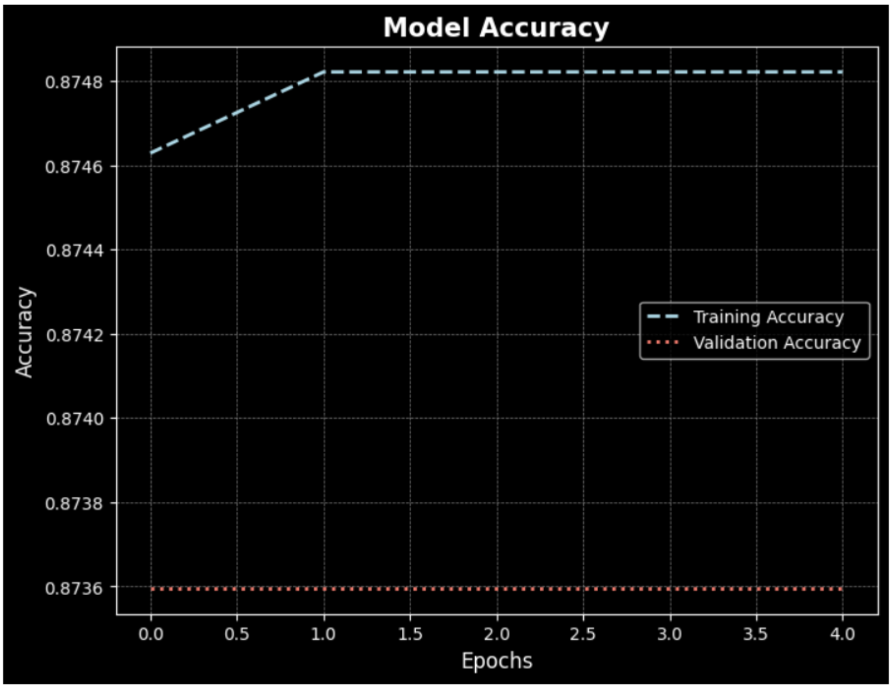


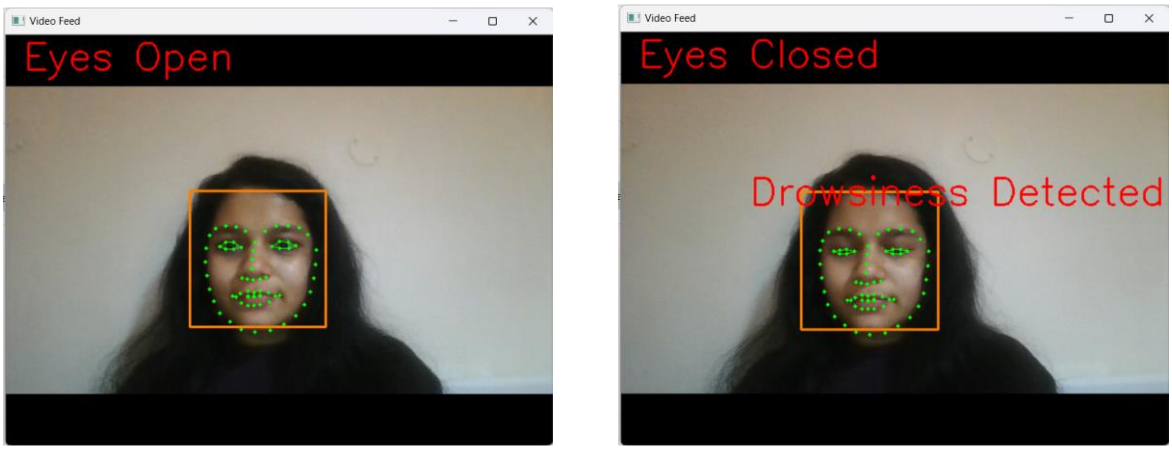
Figure 5. Model's Accuracy Vs Epochs graph

The neural network model's development during five training epochs is documented in the training log that is supplied. The model has a consistent training accuracy of approximately 87.48% across epochs, accompanied by a steady decline in validation loss, suggesting enhanced adaptability to novel data. The best-performing model is preserved by a checkpoint system indicated by the periodic storage of model weights to "Stearing.h5". The general pattern indicates that the model will converge as it adjusts its parameters, bringing it closer to the goal of reducing error and improving precision.

6.4 Discussion

The outcome for the drowsiness detection module provides an extensive evaluation of the system's capabilities when viewed in the context of real-time monitoring with simultaneous autonomous steering wheel control, distance calculation, lane detection, and vehicle detection.

When it comes to the sleepiness detection part, using screenshots of a real-time monitoring camera with both closed and open eyes helps evaluate how well the model determines the driver's level of attentiveness. To reliably classify eye states, the system probably uses image processing methods, maybe convolutional neural networks (CNNs). The precision, recall, and F1 score accuracy metrics—which are reported—offer a numerical assessment of the efficacy of the drowsiness detection module.



A. Eyes Open

B. Eyes Closed

Figure 6. Drowsiness Detection

When it comes to the sleepiness detection part, using screenshots of a real-time monitoring camera with both closed and open eyes helps evaluate how well the model determines the driver's level of attentiveness. To reliably classify eye states, the system probably uses image processing methods, maybe convolutional neural networks (CNNs). The precision, recall, and F1 score accuracy metrics—which are reported—offer a numerical assessment of the efficacy of the drowsiness detection module.

Figure 7. Lane and Vehicle Detection, Distance Estimation

Screenshots of autonomous steering wheel functions, distance computations, and lane identification are included to provide important context for the review. Screenshots of lane detection demonstrate how well the system can recognize and follow traffic lanes. Screenshots of the automated steering wheel show how control techniques based on sensing of the environment are implemented in practice. Determining distances, which frequently include estimating depth, helps to comprehend the system's spatial awareness.

Figure 8. Alert System shown on Drowsiness detection

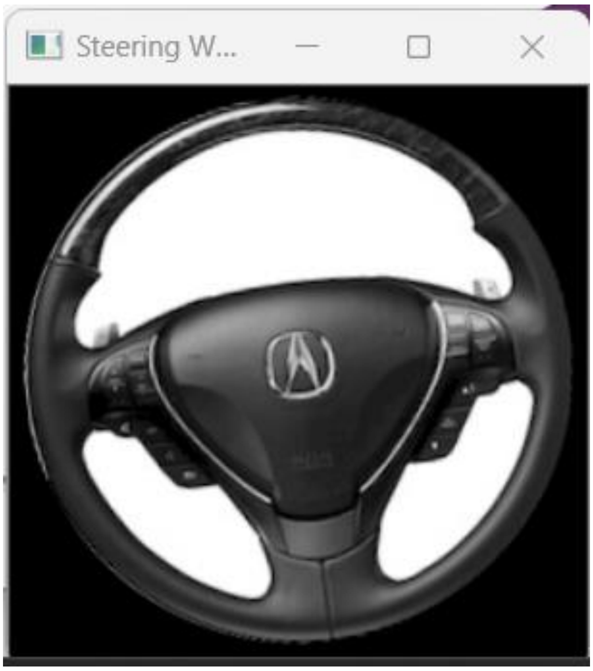


Figure 9. Steering Wheel Automation

In order to effectively communicate these results, emphasize how the many study modules work together to improve driver safety. Examine the difficulties that arose during real-time processing to highlight how flexible the system is to changing traffic conditions. This will provide important information about how robust the research solution put into practice is. The screenshots that go with this text serve as visual proof, demonstrating how well the research project succeeded in creating a thorough and trustworthy driver aid system.

7. Conclusion and Future Work

All things considered, the created driver aid system, which includes automated steering, lane detection, object detection, and sleepiness monitoring, shows promise. Through the integration of various components, crucial factors including vehicle identification, lane tracking, and alerting systems for drowsiness are addressed, offering a comprehensive approach to improving driver safety.

Reactive steering assistance, precise vehicle identification, and efficient lane detection all show that this study has met its objectives. An all-encompassing and proactive safety solution is further enhanced by the distance estimation and alert system.

Although the existing implementation presents noteworthy accomplishments, there are opportunities for further development. Prospective expansions encompass investigating sophisticated object identification models, optimizing the drowsiness monitoring algorithm to accommodate various facial expressions, and integrating real-time meteorological and illumination circumstances to enhance efficacy. The system's capabilities could be further enhanced by incorporating vehicle-to-vehicle communication systems and investigating the possibility of artificial intelligence for adaptive learning in dynamic driving settings. Ongoing cooperation with professionals in the automotive and safety domains can yield significant insights, guaranteeing that the system adapts to new problems in the field of road safety and driver assistance.

References

Z. Wang, J. Zhan, C. Duan, X. Guan, P. Lu and K. Yang, "A Review of Vehicle Detection Techniques for Intelligent Vehicles," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3811-3831, doi: 10.1109/TNNLS.2021.3128968.

V. S. R. Kosuru and A. K. Venkitaraman, "Preventing the False Negatives of Vehicle Object Detection in Autonomous Driving Control Using Clear Object Filter Technique," *2022 Third International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, Bengaluru, India, 2022, pp. 1-6, doi: 10.1109/ICSTCEE56972.2022.10100170.

C. Chen, C. Wang, B. Liu, C. He, L. Cong and S. Wan, "Edge Intelligence Empowered Vehicle Detection and Image Segmentation for Autonomous Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13023-13034, Nov. 2023, doi: 10.1109/TITS.2022.3232153.

B. Xu, B. Wang and Y. Gu, "Vehicle Detection in Aerial Images Using Modified YOLO," *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, Xi'an, China, 2019, pp. 1669-1672, doi: 10.1109/ICCT46805.2019.8947049.

M. Biparva, D. Fernández-Llorca, R. I. Gonzalo and J. K. Tsotsos, "Video Action Recognition for Lane-Change Classification and Prediction of Surrounding Vehicles," in *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 569-578, Sept. 2022, doi: 10.1109/TIV.2022.3164507.

S. Mozaffari, E. Arnold, M. Dianati and S. Fallah, "Early Lane Change Prediction for Automated Driving Systems Using Multi-Task Attention-Based Convolutional Neural Networks," in *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 758-770, Sept. 2022, doi: 10.1109/TIV.2022.3161785.

N. J. Zakaria, M. I. Shapiai, R. A. Ghani, M. N. M. Yassin, M. Z. Ibrahim and N. Wahid, "Lane Detection in Autonomous Vehicles: A Systematic Review," in *IEEE* Access, vol. 11, pp. 3729-3765, 2023, doi: 10.1109/ACCESS.2023.3234442.

Y. Lei, S. L. Phung, A. Bouzerdoum, H. Thanh Le and K. Luu, "Pedestrian Lane Detection for Assistive Navigation of Vision-Impaired People: Survey and Experimental Evaluation," in *IEEE* Access, vol. 10, pp. 101071-101089, 2022, doi: 10.1109/ACCESS.2022.3208128.

D. Ojha, A. Pawar, G. Kasliwal, R. Raut and A. Devkar, "Driver Drowsiness Detection Using Deep Learning," *2023 4th International Conference for Emerging Technology (INCET)*, Belgaum, India, 2023, pp. 1-4, doi: 10.1109/INCET57972.2023.10169941.

P. Khadkikar, R. Banginwar, I. Mandal, U. Matade and M. Mote, "A Comparative Study of Driver Drowsiness Detection Techniques," *2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, pp. 911-915, doi: 10.1109/ICSCSS57650.2023.10169288.

T. S., P. M. L., S. K. and R. Dhanapal, "Literature Review of Drowsiness Identification and Alert System for Real Time Network," *2023 International Conference on Inventive*

Computation Technologies (ICICT), Lalitpur, Nepal, 2023, pp. 754-760, doi: 10.1109/ICICT57646.2023.10134489.

P. S, M. D. M and S. S, "A Review on Drowsiness Prediction System using Deep Learning Approaches," *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, Erode, India, 2022, pp. 1079-1084, doi: 10.1109/ICCMC53470.2022.9753842.

E. Perkins, C. Sitaula, M. Burke and F. Marzbanrad, "Challenges of Driver Drowsiness Prediction: The Remaining Steps to Implementation," in *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1319-1338, Feb. 2023, doi: 10.1109/TIV.2022.3224690.

D. Singh, A. Singh, S. k. Joshi and R. Deorari, "Obstacle Avoidance, Drive Wheel Synchronization, and Line Tracking Capabilities in an Autonomous Robot," *2022 International Conference on Advances in Computing, Communication and Materials (ICACCM)*, Dehradun, India, 2022, pp. 1-9, doi: 10.1109/ICACCM56405.2022.10009273.

Z. Wang, R. Zheng, T. Kaizuka, K. Shimono and K. Nakano, "The Effect of a Haptic Guidance Steering System on Fatigue-Related Driver Behavior," in *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 741-748, Oct. 2017, doi: 10.1109/THMS.2017.2693230.

A. Kumar, K. Stephen and A. S. Sabitha, "A Systematic Review on Sensor Fusion Technology in Autonomous Vehicles," *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2023, pp. 42-48, doi: 10.1109/ICESC57686.2023.10193038.

M. Uma, S. Abirami, M. Ambika, M. Kavitha, S. Sureshkumar and K. R, "A Review on Augmented Reality and YOLO," *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, Trichy, India, 2023, pp. 1025-1030, doi: 10.1109/ICOSEC58147.2023.10275842.

M. Alhafnawi et al., "A Survey of Indoor and Outdoor UAV-Based Target Tracking Systems: Current Status, Challenges, Technologies, and Future Directions," in *IEEE* Access, vol. 11, pp. 68324-68339, 2023, doi: 10.1109/ACCESS.2023.3292302.

A. B. Amjoud and M. Amrouch, "Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review," in *IEEE* Access, vol. 11, pp. 35479-35516, 2023, doi: 10.1109/ACCESS.2023.3266093.

J. Alikhanov and H. Kim, "Online Action Detection in Surveillance Scenarios: A Comprehensive Review and Comparative Study of State-of-the-Art Multi-Object Tracking Methods," in *IEEE* Access, vol. 11, pp. 68079-68092, 2023, doi: 10.1109/ACCESS.2023.3292539.

M. Grujev, A. Milosavljević, A. S. Ilić, M. Ilić and P. Spalević, "Tree Object Detection And Classification Algorithms – Review," *2023 10th International Conference on Electrical, Electronic and Computing Engineering (IcETRAN)*, East Sarajevo, Bosnia and Herzegovina, 2023, pp. 1-6, doi: 10.1109/IcETRAN59631.2023.10192190.

J. g. wang, j. chen, l. b. zhou, w. k. wan, y. w. yau, “Vehicle detection and width estimation in rain by fusing radar and vision,” in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1063–1068.

P. C. Gagan Machaiah and G. Pavithra, "A review article on lane sensing and tracing algorithms for Advanced Driver Assistance Systems," *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2022, pp. 1508-1514, doi: 10.1109/ICCES54183.2022.9835999.

Y. Koroglu and F. Wotawa, "Towards a Review on Simulated ADAS/AD Testing," *2023 IEEE/ACM International Conference on Automation of Software Test (AST)*, Melbourne, Australia, 2023, pp. 112-122, doi: 10.1109/AST58925.2023.00015.