Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4, 1300-1331
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i4.6259

Convolutional neural network-based approach for edge detection in autonomous driving

Marvy Badr Monir Mansour[1*], Ahmed Mohamed Mansour[2]
[1,2]Department of Electrical Engineering, The British University in Egypt, Cairo, Egypt; marvy.badr@bue.edu.eg (M.B.M.M.) ahmed179019@bue.edu.eg (A.M.M.).

Abstract: Autonomous driving systems (ADSs) hold promise for enhancing safety and efficiency on the roads; yet, concerns persist due to rising fatalities involving vehicles equipped with ADSs. This research comprehensively examines the technical components of ADSs, including current challenges, system designs, evolving techniques, and critical features like sensor technologies such as Light Detection and Ranging (LiDAR) and cameras. These sensors enable vehicles to perceive their environment accurately, facilitating tasks such as navigation and obstacle avoidance. Advanced edge detection strategies for lane detection and the usage of Lane Keeping Assist (LKA) structures are crucial technologies for ADS. Hence, in this paper, we implement a modified Sobel edge detection algorithm to improve its performance for lane detection and integrate a CNN-based approach into our system. By trying various Gaussian filter parameters, we develop an optimized edge detection system that performs well in different lighting and weather conditions, such as low light or rainy weather. In our work, we implement a Convolutional Neural Network (CNN) for edge detection and train it using a comprehensive dataset of road images and traffic scenes. The dataset includes a diverse range of conditions, such as different lighting (day and night), weather (clear, rainy, foggy), and road types (highways, urban streets, rural roads). This extensive dataset allows the CNN to learn features robustly and generalize well across various driving scenarios. Simulation and results show that our CNN-based approach has high performance, as it exhibits high accuracy and low processing time needed for ADSs.
Keywords: *Autonomous driving systems, CNN, Edge detection, Lane keeping assist, Machine Learning, PID controller*.

1. Introduction

Automated or autonomous or driving systems (ADSs), have the potential to make driving secure, high-quality, and effective. On the opposite hand, the quantity of fatalities related to automobiles with ADSs is rising. The requirement for improvement in ADS prevents it from accomplishing its complete capability. The technical components of automated riding are thoroughly tested in this research, which additionally conducts in-depth survey on cutting-edge solutions, high-level system designs, growing techniques, and vital functions for self-driving cars or autonomous vehicles (AVs). AVs make use of lots of technological techniques to examine the environment using sensors placed all around the automobile. This consists of sensors which can be used to sense the environment successfully and absolutely, such as Light Detection and Ranging (LiDAR) and cameras. Vision in self-sustaining automobiles is performed with the aid of a number of technologies, which includes sensors. The automobile recognizes and evaluates its environment, such as surrounding vehicles and limitations, while statistics is accrued. Based on these statistics, it then takes the proper moves, in order to get on the meant destination properly and efficaciously. The process of creating appropriate selections is based totally on statistics evaluation that is called planning. Determining the best route Mansour and Said [1] the proper speed and the encompassing environment are all part of the planning, which use auxiliary systems like GPS.

History: Received: 12 February 2025; Revised: 4 April 2025; Accepted: 9 April 2025; Published: 15 April 2025
* Correspondence: marvy.badr@bue.edu.eg

In safety systems, sensors are used in the vehicle's localization process. To improve the localization precision and help the vehicle decide its position, complex algorithms and human-device interface are adopted. Therefore, sophisticated algorithms are essential for quickly and accurately evaluating the vast amount of data collected from sensors and making judgements accordingly. This guarantees prompt decision-making in addition to safe driving for the driver and passengers.

Most ADSs frequently use various sensors and algorithms to break down the complex work of automated driving into less difficult processes. Society of Automotive Engineers (SAE) defines ADS as hardware-software program structures which is capable of supporting dynamic riding obligations in an environmentally friendly way. The SAE categorizes ADS capabilities into five levels of automation, from basic driver assistance to fully autonomous operation, although challenges remain in achieving higher levels of automation due to environmental uncertainties and some human factors. Level one includes the basic driving that involves balance manage, anti-lock braking systems, and adaptive cruise manipulation. Despite concerns, ongoing technological advancements aim to address safety issues and improve public confidence in ADSs, such as safety standards in aviation automation. With advanced technology, emergency braking and coincidence avoidance, become easier. Level-two automation is now a feasible due to partial automation. While, Level three introduces conditional automation, permitting drivers to divert their interest from using regular operations. Yet, they should respond to emergency alerts and be prepared to renew manipulation. Level three automated systems perform inside particular operational layout domain names such as highways. Levels four and five take away the need for human attention absolutely.

However, the fourth stage of self-driving involves the needed infrastructure, on-demand operation and a separate set of maps in case the vehicle leaves its assigned area. Additionally, the automobile is capable of parking itself [2]. In the fifth stage, the vehicle could find its way in any network and in any form of climate. Nevertheless, no car is able to absolutely carry out the self-driving functioning of the fourth and fifth levels. This is because of unexpected environmental conditions, problems introduced by human behaviour. Most ongoing studies aim to create robust self-riding and enhance safety protocols. For independent riding, safety structures involve the protection requirements of automatic aviation, where they incorporate design to lessen hazards and improve performance and safety.

The creation of ADSs represents a modern-day shift in the automobile-driving era. By using various sensors, artificial intelligence, and real-time information processing, ADSs are designed to run on their own. The integration of several sensing technologies, which permit the vehicle to precisely sense its environment, is crucial to the functioning of ADSs. These technologies include cameras, radar, ultrasonic sensors, LiDAR, and others. The potential of the system to discover and examine its surroundings is enhanced through statistics that each sensor gives, for example, LiDAR makes use of laser pulses to supply high-precision 3-dimensional maps of the region across the automobile. This feature is crucial for identifying specific items and mapping the surroundings, which permit the automobile to manoeuvre throughout hard environments. By measuring the gap and velocity of nearby items, radar sensors supplement LiDAR. This is specifically beneficial in exclusive climate scenarios in which optical sensors may not function appropriately.

Ultrasonic sensors assist in parking and obstacle avoidance because they are utilized for short-range sensing. In order to detect lane lines, road signs and vehicle cameras provide a thorough picture of the environment, also data from many sources are combined using sensor fusion algorithms. This integrated approach provides accurate and reliable system's perception. Computer Vision and Machine Learning (ML), especially deep learning (DL), have become increasingly important in interpreting the amounts of data generated by these sensors. Convolutional neural networks (CNNs) are widely used for their visual control capabilities to process information and identify complex patterns. These networks are trained on big data to identify features such as road markings, curbs and obstacles, and contribute significantly to traffic situation awareness. Once the environment is sensed, the ADS uses sophisticated planning algorithms to predict the behaviour of the vehicle. This application involves path planning, which includes calculating the optimal path, and speed planning, which predicts the vehicle speed and direction. This model should account for dynamic objects such as moving vehicles, pedestrians and unexpected obstacles to ensure safe and efficient travel.

An important feature of ADSs is edge detection, especially when it comes to lane detection, obstacle detection, and scene detection. Traditional edge detection techniques such as Sobel, Canny, and Laplacian filters have been widely used for ease of use. Computational efforts fail in unpredictable and complex driving situations, such as changing lighting, bad weather and difficult roads. Because CNNs can learn hierarchical features and patterns from large datasets, they offer great improvements in many ways. CNNs are well-suited for complex edge detection tasks because they can learn and extract features from unprocessed image. The recall metric assesses the ability of the model to detect any significant edges, while the precision metric assesses the accuracy of those reported edges. Processing speed is important for real-time applications because ADSs need to detect edges quickly and efficiently while making effective decisions compared to traditional edge detection methods.

1.1. Anticipated Problems

Self-driving aims to provide more comfort and convenient driving experience. Specifically, it attempts to reduce traffic accidents caused by human negligence of elements like lane changing, road dashing, and lack of recognition, all of which might be of substantial concerns. Research and development of ADS works on the enhancement of traffic flow, safety within roads, and reliable transportation options. This is vital since accidents, mainly due to human mistakes, serve as one of the main causes of injury and loss of lives across the globe. ADSs drastically contribute to fewer injuries by reducing human errors via sensors usage, Artificial Intelligence (AI), and ML algorithms. These systems make real-time selections to enhance driving protection and performance.

Essential technology inside ADSs is CNN, which processes large amounts of visual statistics from cameras and sensors that are embedded on self-riding vehicles. CNNs could detect patterns, such as lane markings, street signs and boundaries. They are used in detecting lanes successfully for the vehicle to observe its correct lane. This functionality is crucial for self-driving cars to function in various and uncertain environments. On the other hand, the potential of AVs in logistics are expected to promote supply chains, delivery times, and decrease the overall operational costs for more efficient and sustainable business practices.

However, the development of ADS is complex, and the uncertainty of the real-world environment brings many challenges related to reliability, safety, ethical issues, legal issues, and public trust. ML and CNN play a core role in promoting the adoption of ADS by enhancing the accuracy, reliability, and safety of ADS. The essential purpose of our research is to improve ADS where CNN and ML are used to design accurate lane recognition system. Our target is improving road safety and minimizing mistakes of drivers that may cause deviations of a vehicle from lane and hence reducing traffic accidents. This work's goal is to utilize CNN to solve the previous problems and enhance the traffic performance. Additionally, this work is aimed at improving road networks while increasing accessibility for people with mobility limitations through ML strategies for better performance in different conditions. Thus, bringing about solutions related to issues of accessibility, traffic congestion, and road safety.

1.2. Paper Objectives and Contributions

The primary objective of this research on autonomous cars is to explore and create a novel way to improve the vehicles' ability to adapt in unforeseen situations. This means concentrating on reducing obstacles associated with making judgements in real-time, maintaining uninterrupted communication, and integrating sensors to continuously produce precise data for correct analysis prior to making decisions. In addition, the research aims to solve human error problems and advance ADS culture in a safe and dependable manner.

In addition, by identifying the best route, this study seeks to investigate other possible social advantages that could enhance both people and vehicle traffic safety and ease congestion. Serving people with mobility impairments is one of the main goals in the development of ADS technology. This plays a crucial role in people's safety and facilitates their transportation between locations without the difficulty of driving. To accomplish these goals, a CNN is employed for autonomous car edge detection in this study. One of the core tasks of Computer Vision is edge detection, which is locating the borders of images. Accurate edge detection is essential for activities like lane detection, obstacle recognition, and general picture interpretation in ADS.

By using CNNs for this purpose, the vehicle's capacity to make deft decisions in real-time is enhanced by utilizing the capabilities of ML to process and analyse large volumes of visual input. Because driving situations in the real world are unpredictable, a system that is flexible enough to adjust to unexpected occurrences is required. Conventional rule-based systems frequently fall short in taking into consideration the multitude of variables found in typical driving situations.

The goal of the research is to enhance the automobile's dynamic version by using CNNs because CNNs can recognize complicated styles in visual inputs. Higher level decision-making algorithms that incorporate information from other sensors like GPS, radar, and LiDAR use the CNN's edge detection outputs. The integration of data guarantees a comprehensive perspective of the surroundings, augmenting the car's situational awareness. Steady communication between these components is made possible by a strong architecture that has high reliability and low latency. This architecture makes sure that vital data is processed and transferred quickly, enabling the autonomous system to keep a consistent awareness of its environment and make correct judgements in the moment.

The key contributions in this research include advanced edge detection strategies for lane detection and usage of Lane Keeping Assist (LKA) structures, which are essential technologies for self-driving. This paper's main contributions are as follows:

- Advanced Edge Detection Implementation: We have implemented and modified the Sobel edge detection algorithm to improve its performance for lane detection under different lighting and weather conditions. Through trying various Gaussian filter parameters, we have developed an optimized edge detection system that performs well in difficult conditions such as low light or rainy weather.
- Integration of ML and CNNs: The combination of CNNs and ML methods for path recognition is an important fundamental in our research. By building and training a CNN to detect trails in images, we were able to significantly increase the accuracy and reliability of trail detection methods. Our system can learn from big data and adapt to different driving situations using ML techniques.
- Enhanced LKA Systems: We improved the vehicle's ability to detect and track lanes with LKA systems by adding advanced CNN-based lane detection and edge detection techniques. Because all the ADSs in this combination are more reliable and safer, the vehicle can set its course with the same accuracy.
- Performance Gains: Compared with current methods, the proposed methods show a noticeable increase in accuracy and flexibility. Our CNN-based path finding method is efficient and effective for real-time applications, resulting in lower processing time and increase in detection accuracy.
- Real-World Applicability: We evaluated our strategies in simulated situations that intently reflect real-world driving situations to compare their applicability. Our method tested terrific upgrades in lane recognition precision in inclement weather and at night-time, demonstrating its usefulness.
- Impact on Technology: This study results in the creation of autonomous motors which are more reliable and effective. We establish a basis for destiny studies and industrial programs centred at enhancing the safety and effectiveness of self-driving technologies by improving edge detection and adopting CNNs and ML along with LKA structures.
- Future Research and Possible Directions: Our techniques are scalable and versatile enough to be applied to other areas of ADS, like obstacle avoidance and pedestrian recognition. By further

combining ML learning methods with edge detection, future research can expand on our findings and improve performance and flexibility under a variety of driving scenarios.

- Social Impact Assessment: Analysing how improved ADS technologies are being adopted. One benefit of these enhanced automated technologies is a decrease in traffic accidents by guaranteeing that cars follow posted speed limits and reducing traffic yielding to less congested routes. Moreover, enhancing the mobility of people with physical limitations is a major factor. All these advancements shorten the learning curve, boost the benefits of ADS overall, and save time. ADS advances through ongoing research, which increases the adaptability, resilience, and capacity of vehicle systems to handle difficulties in unforeseen circumstances.

This paper constitutes of the following sections. Section 2 provides a brief background about our work. Then, Section 3 discusses usage of Computer Vision and ML for AVs. While Sections 4 and 5 describe the state-of-the-art literature and the adopted research methodology respectively. After that, Section 6 explains the simulation carried out in our work and the obtained results. Whereas, Section 7 compares our work with other existing approaches. Finally, we conclude this paper and provide some future insights in Section 8.

2. Background

2.1. System Architecture

The categorization of ADS architectures is illustrated in Figure 1 [3]. ADSs are formulated in either standalone configurations, functioning exclusively for the host vehicle (ego-only systems), or as interconnected multi-agent systems. Additionally, these design principles are implemented through two distinct approaches: modular and end-to-end driving approaches.

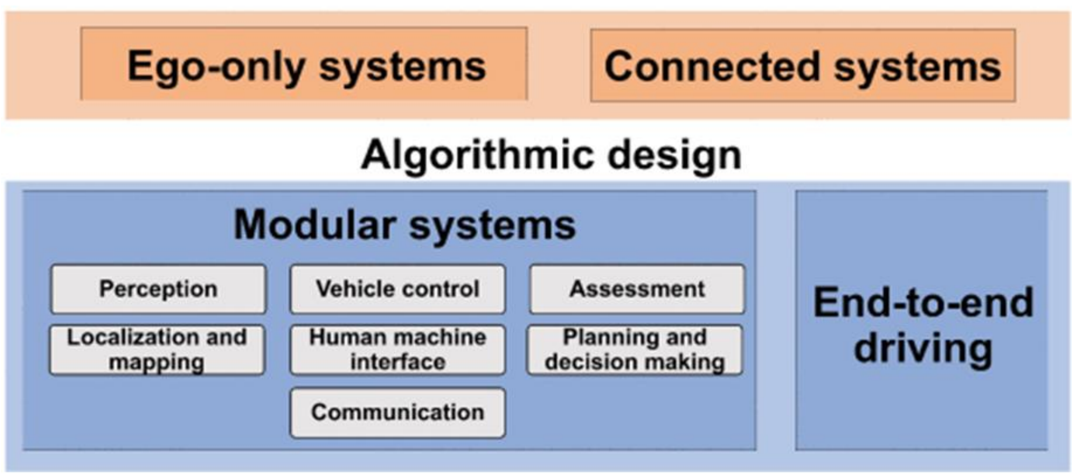


Figure 1.
High-level automated driving system architectures.

2.1.1. Ego-only System

The execution of driving's basic functions is essential. The vehicle's automated driving, which may depend on other cars and infrastructural components, is one of the necessary functions. But strategy is not the only thing that the ADS relies on. It is a networked ADS, and most of its peers use the self-approach. This is mostly because of how ADSs are developed and the extra complexity that comes with linked systems [4].

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

2.1.2. Modular Systems

Modular systems comprise a series of independent parts, each linking actuator outputs to sensory inputs. Perception, evaluation, planning and decision-making, mapping and localization, vehicle control, and human-machine interface form are the core functionalities of modular ADS. Typical modular pipelines start with raw sensor data fed into the localization and object detection modules and then progress to scene prediction and the decision-making phases. Separate development is enabled by breaking down this complex task of automated driving into modules, drawing on existing knowledge from robotics, Computer Vision, and vehicle dynamics. The real benefits of modular systems are related to the transferable experience in which algorithms and functions can develop and be integrated into such a modular framework. For example, safety can be put on top of an advanced planning module to enforce specific emergency procedures without modifying the architecture's internal operations, which facilitates the development of redundant yet reliable architectures. However, over-complexity and the possibility of error propagation are significant drawbacks of modular systems.

2.1.3. End-to-end Driving

End-to-end driving is directly derived from the sensors and detectors through drive and direct perception. Continuously monitoring the steering wheel and pedals are required, with an emphasis on accelerating the car and slowing it during turns, managing turns in ego-motion. DL and AI are essential to this form of driving since they thwart direct monitoring of the vehicle and used in deep reinforcement learning (RL). Based on the requirements, limitations, and mission of ADS, all driving systems have so far demonstrated disadvantages alongside the benefits [5].

2.2. ADS Implementation

The integration of ADS through modular systems comprises several essential elements [6]:

- Plans for Sensors: Plans for sensors have a unique set of advantages and disadvantages. The selection of sensor arrangement is contingent upon a number of aspects related to the capabilities of the vehicle.
- Software Architecture: The software architecture of an autonomous vehicle (AV) plays a critical role in enhancing its perception and planning capabilities, often improved through Simultaneous Localization and Mapping (SLAM).
- Data Acquisition: Data acquisition is a crucial aspect of autonomous systems, with data collected from various sensors utilized for environment perception and decision-making.
- Perception Algorithms: Utilized to process data obtained from sensors, perception algorithms are instrumental in comprehending the surrounding environment.
- Vehicle Interfacing: This involves the interaction between the autonomous system and the hardware of the vehicle.
- Failsafe and Fail-Operational Functionalities: These functionalities are integral for ensuring the safety of the passengers and reliability of the ADS architecture.
- DL Applications: The growth of using DL applications in develop ADS forms the foundation for scene perception, path planning, and algorithm behaviour regulation.
- Model Based Safety Analysis (MBSA): When used on an Advanced Driver Assistance System (ADAS), Modular Numerical Simulation is utilized.
- System Modularity: It enhances fault-tolerant characteristics and reduces computational complexity.
- Other components include**:** the sense component, which lets the AV perceive its environment; the deciding component, making decisions based on information gathered via sensors; and act component, acting on that decision as soon as possible.

All the components mentioned above should be correctly implemented, safely developed, and tested to a large extent to guarantee the safety of the passengers and the vehicle's reliability. In a networked ADSs, there are two major issues related to:

- Requirements for Connectivity: In order to fully utilize their potential, autonomous cars need to have improved connectivity. High-performance clusters are arranged inside functional domains using complex, real-time architectures, and they are connected by a central gateway as part of a high-speed data backbone system. High-speed data will be more and more needed as driverless cars integrate complex systems that produce large amounts of data.
- Cloud-Connected Systems: A cloud-connected ADS takes into account aspects as network delay, fault tolerance, and network security, ensuring robust and secure operation [7, 8].
- Multidisciplinary Approach: It, therefore, entails autonomous and networked systems involving a multidisciplinary team consisting of specialists in various domains related to ML, AI, information security, study, and technology development in making informed decisions. It will also comprise policy making and innovation of advanced sensing technologies involving human aspects.

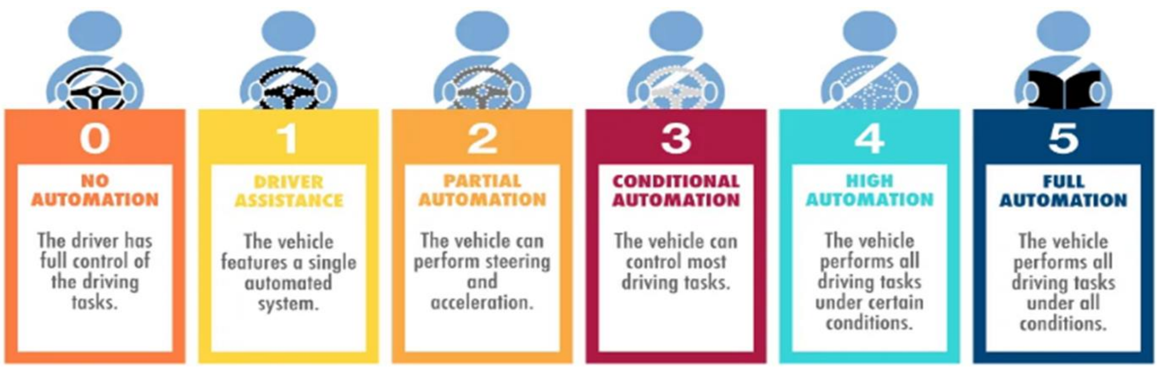


Figure 2.
Levels of ADS.

- **Value Creation:** The advent of ADS holds the potential for significant value creation for drivers, the automotive industry, and society at large. It has the capacity to enhance safety, convenience, and utilization of time on the road, benefiting various demographics, including elderly drivers.
- **Future Prospects:** Industry leaders need to grasp connection in order to provide the promised vehicle-to-everything (V2X) capabilities of completely ADS, even though the future of AVs is still unclear.

2.2.1. ADS Simulation Platform

Datasets form an essential part of ADS that are used by ADS algorithms since those need training and testing. This is done by passing through sensory data in various algorithms, all having an intended goal that is often measured on annotated datasets. Primary building blocks, such as object detection and tracking, are standard, while other fields like Computer Vision have dedicated annotated datasets aiding development [6].

The early instances, such as the [PASCAL VOC] dataset and [KITTI] Vision Benchmark, were followed by more representative datasets, of which [KITTI] still provides a better reflection of driving scenarios. While ImageNet and COCO are suitable for training, they miss context in their image labels to be representative for testing ADS. Other notables include UC Berkeley Deep Drive, Oxford Robot Car, Cityscapes, Toronto City, nuScenes, Comma.ai, DDD17, LiVi-Set, and Common Road. Each dataset has its strengths, and thus, some give sensor information like LiDAR, GPS, and image sensors. SHRP2, 100-Car study, euro-FOT, and NU-Drive are some examples of naturalistic driving datasets for driver behaviour knowledge.

Open-source frameworks can bridge business and research entities in immensely boosting the development of ADSs. Some notable open-sourced frameworks are effective in implementing ADS platforms in real-world scenarios, promoting democratization in the development, are Autoware, Apollo, Nvidia Drive Works, and open pilot.

In ADS simulation, car instrumentation is replaced with expensive experiments that overcome some constraints needed for road testing. Simulation frameworks like CARLA focused on urban driving [9]; TORCS, racecourse simulation; or Gazebo, oriented to robotics, serve as a base where algorithms can be tested before modules are applied onto the road. It will, moreover, be easy to replicate dangerous situations like collisions with pedestrians and investigate them.

2.2.2. ADS Levels

Several organizations, including German Association of Automotive Industry (VDA) and Society of Automotive Engineers (SAE), have categorized ADS into distinct stages. There are five levels of automation shown in Figure 2 [10]: Level 0 (no automation) to Level 5 (complete automation). The car has no automatic features at Level 0.

Automatic braking and other driver assistance features are introduced at Level 1. The car can accelerate and brake at Level 2, but the driver has to stay alert and prepared to take over at any time. Under some circumstances, Level 3 cars can function independently, allowing the driver a little window of time to take back control if needed. The High levels of automation are possible with Level 4, allowing the car to operate autonomously in many driving situations. At Level 5, the car is fully automated, able to handle every part of driving under any conditions.

Commercially available AVs have not yet achieved automation levels higher than Level 2 or Level 3. Leading the charge in the development and testing of AVs, automakers Tesla, have been approved by regulatory organizations in a number of locations, including Nevada in the United States and several European nations, subject to certain restrictions. Modern developments in AVs have their roots in the early experimental models, including those created by Carnegie Mellon University in the 1990s. Autonomous taxis are being developed and tested by ride-sharing services such as Uber and Lyft. Singapore concluded an autonomous taxi test in 2018 [11]. Numerous safety advantages could result from autonomous cars, major among them the prevention of accidents Guo, et al. [12] and death caused by human error — human error accounts for 90–95% of all car accidents.

2.3. Current Situation

The integration of data from several sensors, including radar, LiDAR, and cameras, enables sensor fusion techniques to create a full picture of the environment around the vehicle. This integration makes driving in difficult situations less likely to go incorrect and increases overall dependability. But since AVs produce enormous amounts of data that could be hacked and misused, data security is still a major issue. The development and implementation of AVs should carefully examine ethical, legal, and technological difficulties, even though they hold great promise for enhancing road safety and efficiency [13, 14]. It is important to assure the reliability and security of AVs. For example, predictive analytics can detect irregularities in sensor data and predict traffic patterns, enabling AVs to develop countermeasures to prevent collisions.

Furthermore, AI systems assure that if a system fails, others can compensate, maintaining overall vehicle safety. For AV technology to advance and be implemented safely and effectively, it is imperative that ML, Computer Vision, and sensor fusion techniques be integrated [15]. Due to developments in ML, AVs can now identify trends and make judgements in real-time by utilizing up-to-date data. CNNs and other DL methods enable AVs to quickly adjust to new conditions by learning from historical data [16]. Comparably, Computer Vision technologies are essential to AVs' ability to "see" and comprehend their surroundings more fully. These systems analyse photos and videos taken by the car's cameras to recognize traffic signs, people, barriers, and other important factors that influence driving decisions.

3. Methodologies for AVs

Advancements in Computer Vision is essential to the development of ADSs. CNNs, in particular, are essential to ML because they enable the recognition and interpretation of visual data from the environment around the vehicle. To reliably identify lane markings, traffic signs, pedestrians, and other impediments, CNNs are trained on big datasets.

Furthermore, the vehicle's capacity to identify lane edges and other essential features is improved by the application of edge detection techniques like the Sobel operator. Gaussian filtering ensures robustness in diverse contexts by significantly enhancing edge detection efficiency across a range of situations.

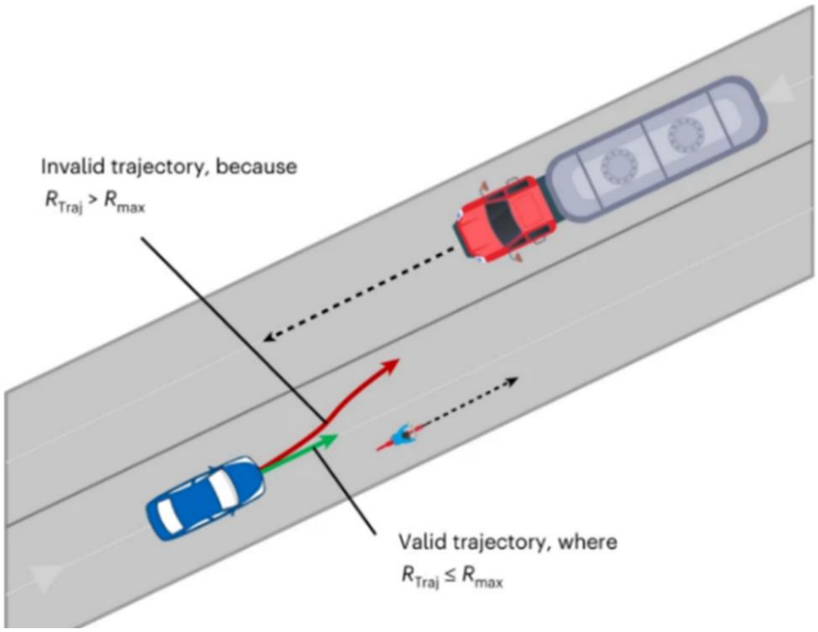


Figure 3.
Choosing the suitable trajectory.

Path prediction and ML techniques have made significant progress in the field of AVs. With a particular focus on these technologies usage in AVs, this study identifies important themes and current advancements in research.

3.1. Path Prediction

ML plays an important role in improving AVs prediction as traditional methods have been significantly enhanced by ML techniques. Especially DL-based techniques, which make use of enormous datasets and complex algorithms, have demonstrated impressive performance in forecasting vehicle paths. For example, some studies offer an overview of motion prediction and risk assessment methods utilized for AVs.

DL-based car behaviour prediction has become more popular in the last several years. These methods improve vehicle behaviour analysis. Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks are examples of sequential models that have been used extensively to capture temporal dependencies in trajectory data. Time-series data processing, which is essential for precise trajectory prediction in dynamic driving settings, is a specialty of these models [17].

3.2. Trajectory Prediction

Trajectory Prediction is the process of projecting moving objects' future positions from their historical and current conditions as shown in Figure 3 [18]. This can be achieved by several methods, such as DL techniques, statistical models, and physics-based models. While probabilistic models, such as Gaussian Processes and Hidden Markov Models, incorporate uncertainty into their predictions, physics-based models use laws of motion to forecast future positions. These predictions are further refined by DL techniques via use of neural networks to learn from vast datasets.

CNNs have successfully improved Trajectory Prediction and also have dramatically changed the field of Computer Vision. They work effectively on jobs involving spatial data, such pictures and videos. CNNs are used to analyse complex images and predict the future paths of objects inside those scenes by utilizing their capacity to automatically detect and learn spatial hierarchies from raw pixel data.

Furthermore, predicting human movements and interactions with vehicles is one prominent use of CNNs in AVs. In order to forecast pedestrian trajectories in urban contexts, for instance, some works suggested a CNN-based framework that examines the spatial interactions between pedestrians and cars.

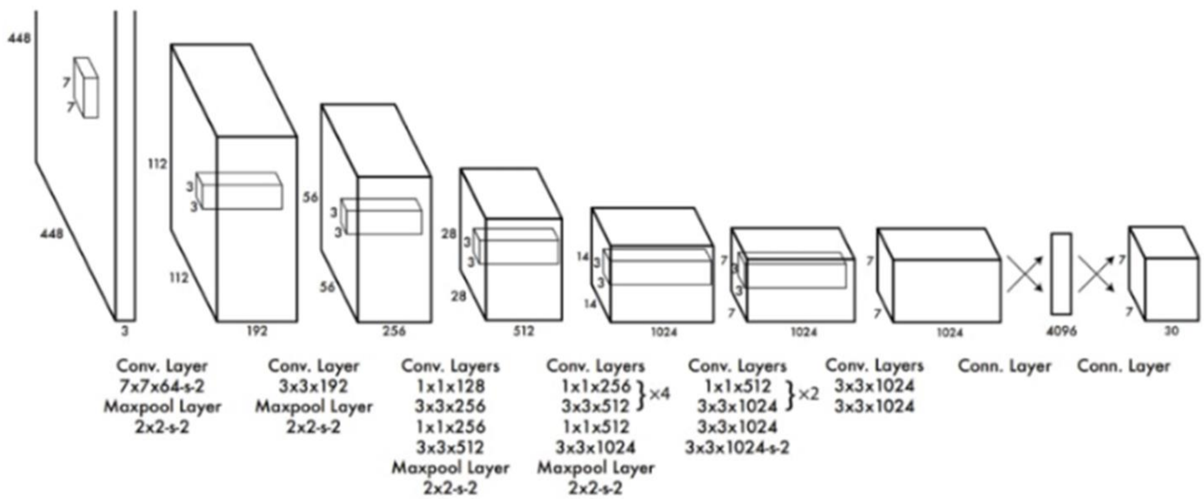


Figure 4.
CNN Architecture Framework.

This method makes use of the convolutional layers to extract high-level features from image data, enabling the model to understand intricate details of pedestrian behaviour and make accurate predictions. Different traffic participants' interactions and geographical linkages are captured using CNNs and Graph Neural Networks (GNNs). These techniques work especially well in urban settings because there are a lot of moving parts and intricate interactions between cars, customers, and other objects.

3.3. Object Detection and Scene Perception

Developments in DL, ML, and AI have played a major role in the development of AVs. Object detection is a vital component of AV technology that helps the car recognize its surroundings. The capacity of CNNs to learn and extract hierarchical features from visual data has made them an essential component for this goal. AVs need object detection in order to identify and locate different items in their environment. Three stages make up the procedure in general: feature extraction, region proposal, and classification.

3.3.1. Region Proposal/Region Selection

To scan the full image at several scales, region proposal methods such as window sliding were initially used. Unfortunately, this approach requires a lot of processing power and so is not appropriate for real-time use in AVs. By directly producing region proposals from the convolutional feature maps, contemporary methods like Region Proposal Networks (RPN), which are used in Faster Region-based CNN (R-CNN), have increased efficiency.

Conventional techniques for feature extraction such as the Haar-transform and Histograms of Oriented Gradients have become popular but not very resilient to changing environmental conditions. By deriving spatial hierarchies of features from raw pixel data, CNN algorithms have transformed feature extraction and increased adaptability to the dynamic situations that AVs encounter [19].

3.3.2. Classification

ML methods are used to classify things after features have been extracted. Prior to the development of DL, methods like Support Vector Machines (SVM) and the Deformable Parts Model (DPM) were frequently employed. These days, CNN-based models are more popular because of their better performance and ability to learn from start to finish.

3.3.3. Object Detection

In Computer Vision, CNNs have proven essential, especially for object recognition in AVs and image categorization. Multiple layers make up CNNs as shown in Figure 4 Ammar, et al. [20] which perform feature extraction by using supervised learning to acquire increasingly complicated features. Advanced CNN architectures, including Very Deep CNN (VGGNet), Dense CNN (DenseNet), and Residual CNN (ResNet), have been used to improve accuracy of object detection in AVs.

Well-known DL models for real-time object identification include You Only Look Once (YOLO) and Single Shot Multibox Detection (SSD). By adding more convolutional layers to intermediate layers of a pre-trained network, SSD efficiently addresses the issue of scale fluctuations and combines object detection tasks into a single network. To achieve a balance between speed and accuracy, YOLO splits the image into a grid and forecasts bounding boxes and class probabilities for each grid cell.

LSTM networks are one type of RNN that has been used for sequence prediction tasks like visual tracking in AVs. By preserving a recollection of prior time steps, these models are able to capture temporal relationships, which is essential for forecasting the trajectory of moving objects [21].

Other DL architectures found in AVs are Deep Belief Networks (DBNs) and Stacked Auto Encoders (SAEs). DBNs are made up of several layers with latent, stochastic variables that are greedily learned layer by layer. Although they are less popular than CNNs for object detection, they are efficient at learning high-dimensional representations. The ability of SAEs, on the other hand, to encode input data into a lower-dimensional representation through unsupervised learning makes them useful for feature extraction in AV vision systems. However, unsupervised learning and SAEs are out of scope of this paper.

4. Literature Review

4.1. Existing Methodologies

The existing methodologies include avoidance of occlusions, changes in light, and adverse weather conditions. DL models will also need continued improvement and integration with multimodal sensor data to produce human perception and cognition capabilities in an AV. Some of these methodologies are discussed below as follows:

- Occlusions and Partial Visibility: Occlusions, where an object is partially or fully hidden by another object, is one of the biggest problems with object detection for AVs. To make them more secure, CNNs need to be trained on datasets that have occluded objects. Bad Weather, such as snow, fog and rain can disable AV sensors and the DL models that come with them. Using sensor

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

fusion methods which combines data from LiDAR, radar and cameras can help develop a more robust perception system [22]. Future research should focus on making the DL models more robust and generalizable to different driving scenarios. Transfer learning has shown to improve object detection by fine tuning pre-trained models on AVs datasets.

- Introducing AI and ML for AVs: The most significant transformation of present transport systems is the adoption of AI and ML in different forms of AVs. It is remarkable when discussing AVs that ML in general and DL techniques specifically have turned out to be critical to the formation and activity of AVs. The purpose of these technologies will focus on the role and the importance of the technologies, the challenges, and the opportunities that exist in the ADS field. Since the fundamental concept of AVs is based on the vehicle's capability to drive the car and sense the environment, these technologies enhance the efficiency and safety of the car [23].

Perception and Awareness of the Environment: These are among the most paramount functions utilized by the AVs in that it puts them in a position to understand as well as comprehend features of the environment they are surrounded by. That involves usage of devices such as LiDAR sensors, radar sensors, and cameras that capture objects and data, data which undergoes filtering by AI algorithms. Here, the importance of CNNs means clear shape and does good work in such aspects as item identification and detection. In images and videos analysis, CNNs are helpful; they could be applied to such activities as recognizing traffic signs which have different features of a different nature and lay lines on the road, and analysing other moving objects in the traffic.

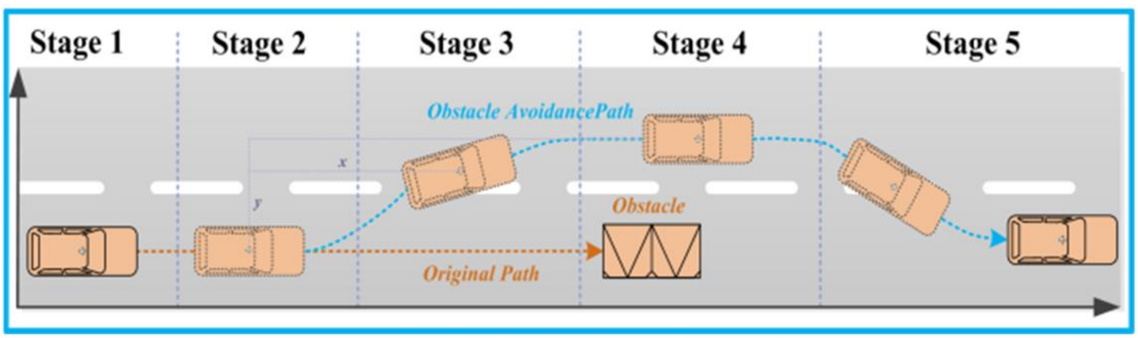


Figure 5.
Path planning and decision making.

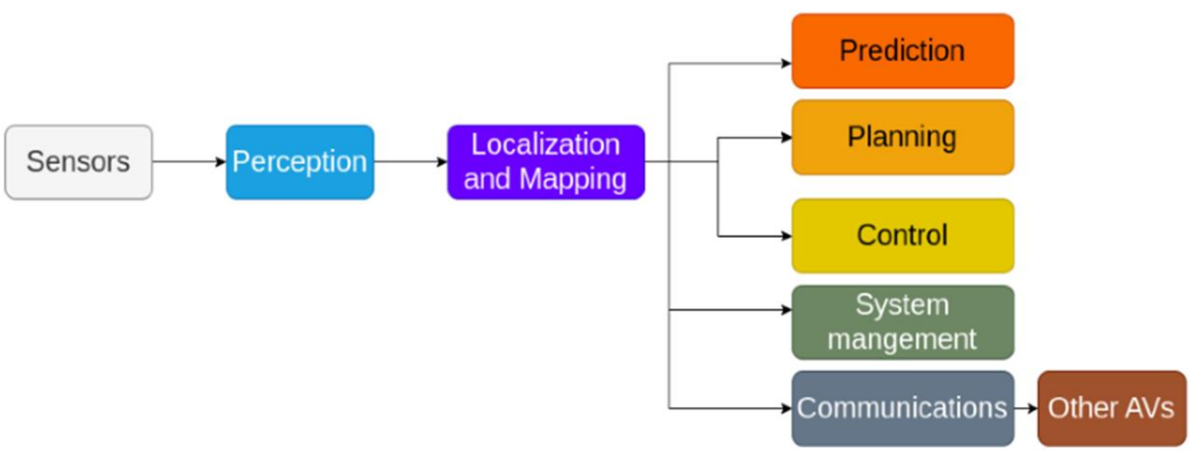


Figure 6.
Mapping and localization.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

Environmental perception in the identified domain of AVs consists of object recognition of pedestrians and understanding of intricate traffic scenarios. Other more advanced structures of DL used in the modelling of the real world include RNNs that are used in the simulation, and Generative adversarial networks (GANs) that are used in the prediction of movements by the walker that challenge the AV systems as well as the generation of possible scenarios in a real-world environment.

- Path Planning and Decision-making: AVs benefit greatly from AI in another important area. The terms decision-making, organizing, commanding and coordinating are used in a significant manner when it comes to human resource management. AV driving decisions that allow for decision-making in real-time and the provision of judgments needed for safe operation depend on ML as depicted in Figure 5 [24].
- Local Path Planning: It determines the best motion of the car and its path to go through. AI techniques are usually used in solving such problems using Rapidly Exploring Random Trees and A* search. These algorithms aid in the generation of available paths that ensures an entity goes through without hindrances or barriers. In parallel, some methods of sensor fusion provide the construction of information from different sensors, improving the reliability and accuracy of the route.
- Mapping and Localization: Accurate mapping and local navigation of AVs, depicted in Figure 6, are essential. High-definition maps blended with AI algorithms permit AVs to recognize their environment with high precision, facilitating higher navigation and obstacle avoidance [14]. ML models use AI-powered techniques such as SLAM to create and update maps in real-time by tracking vehicle conditions within these maps.

4.2. Current Technologies

One essential element of ADS is LKA systems, which work to improve road safety by reducing accidental lane departures. A comprehensive overview of ADS is given in Yurtsever, et al. [3] which emphasized the need for reliability and the possible social effects of wide adoption. The development of ADS technology, system architectures, hardware, sensors, and the challenges of driving automation are discussed as follows:

- Perception and Sensor Fusion: investigate how perception tasks in ADS can benefit from the combination of LiDAR and video data. Though multi-modal sensor fusion is their main focus, the concepts also apply to LKAs, which use camera data for lane detection. Although there is a crucial need of multi-modal integration, camera-based perception is prioritized in the context of LKAs to reliably recognize lane markers.
- 3D Object Detection and Semantic Segmentation: concentrate on 3D detection in ADS, emphasizing its use in path planning, collision avoidance, and motion prediction. Accurate lane detection is the main priority for LKAs, even though 3D detection is essential for overall vehicle safety. Camera-based image-based techniques are crucial for lane recognition boundaries and other road features. There are challenges in aligning semantic information across different modalities needed when integrating camera data with other sensor inputs.
- LKA using PID Controller and PID Management: In LKAs, a proportional-integral-derivative (PID) controller is usually used to keep the vehicle on a specified path. The PID controller adjusts the steering angle based on the difference between current vehicle position and desired navigation position. This approach keeps the car on track and provides smooth, fluid handling. PID controller is a highly desirable product for LKAs due to its efficiency and simplicity.
- Adaptability and Performance: PID controllers work well in straightforward, predictable driving situations, but they are not as good in more dynamic ones. They might be sensitive to noise and disturbances and need to be precisely tuned to accommodate different road conditions. PID controllers are resilient when correctly adjusted, despite these drawbacks, making them appropriate for many LKAs applications.

ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

- CNN for LKAs: CNN functions in lane detection: CNNs are needed for the purpose of processing and evaluating images from cameras in LKAs. Also, they are trained on large datasets to identify patterns and features such as road markings, road signs and obstacles. This assures that the vehicle will remain on the intended course by providing accurate road visibility. CNNs provide a reliable classification and extraction method, so increasing the reliability of LKAs.
- Edge Detection and Feature Extraction: Edge detection techniques—such as the Sobel operator—are widely used in CNN to improve the ability of a vehicle to detect road edges. These techniques enhance the robustness and recognition of CNNs under different conditions. By improving the capture of objects from camera images, edge detection helps CNN detect lane boundaries more accurately, contributing to the overall effectiveness of LKAs [25].
- CNN and PID Controller: can be integrated to take benefit of the advantages of both systems. PID controllers provide quick and smooth steering modifications, while CNNs enable reliable and accurate lane detection. Because of this integration, LKAs are guaranteed to function well in a variety of driving scenarios, so maintaining lane position correctly and consistently. A complete LKA solution is produced by combining CNNs for perception along with PID controllers.

4.3. State-of-the-Art Approaches

The authors in Mao, et al. [26] provided an extended technical background on ADSs challenges and promises. They elaborated on the robustness demanded by ADS and the possible societal impact in case a massive number of vehicles exist. In their research, they mentioned the historical backdrop of automated driving, covering important works that have been ongoing for decades and shaped today's state-of-the-art ADS technology. The paper presented system architectures, hardware, and sensors and discussed driving automation complexity. It further described various components and how ADSs are designed. Some problems were discussed that arise from dark scenes, objects that look different, also how methods of 3D object recognition were mentioned as a solution to these problems. The paper discussed perceptual tasks in ADS, mainly 3D object detection and semantic segmentation related to image-based object detection. They described the development of perception algorithms such as camera-based perception, and event camera-based vision. Furthermore, perception-related issues like lighting change and development in 3D were presented.

The survey in Huang, et al. [27] examined LiDAR and video data perception tasks in ADS. The approach catered to problems resulting from noisy raw data, underutilized information, and mismatch of multi-modal sensor data. In Huang, et al. [27] the fusion techniques were categorized into two significant categories which are Weak and Strong Fusion, each further divided into four smaller groups. Also, the research provided an in-depth review of more than fifty relevant studies, and classified them by fusion stages then reviewed them against open issues and future research directions. The study noted the limitation of single-modal data; and how LiDAR technology and camera data complement each other in a way that results in multi-modal integration; highlighted perception tasks in ADS. They authors in Huang, et al. [27] also included some popular open test datasets for ADS perception tasks such as Waymo, KITTI, and nuScenes. They gave representations of LiDAR and image data. Point-based, voxel-based, and 2D mapping-based point cloud representations were presented alongside fusion methodologies, such as early-fusion, deep-fusion, late-fusion, and asymmetry-fusion. Experimental results on the KITTI test dataset for bird's eye view (BEV) and 3D object detection tasks were discussed and performance of various fusion methods were compared too.

In Caesar [28] the authors presented nuScenes dataset as the first holistic dataset of all car sensors, including camera, radar, and LiDAR, and contains one thousand scenes of twenty seconds each. Each dataset has annotations for 3D bounding boxes for twenty-three classes and eight attributes with precise metrics introduced for 3D tracking and detection. They pointed out the importance of multimodal datasets by citing the fact that no single sensor type is sufficient, and they are complementarity. The study investigated the difficulties in the construction of perception systems regarding ADS and provided emphasis on the need for benchmark datasets for training and evaluation of ML methods. Also, the presented nuScenes dataset, which is large in terms of size and complexity, provides a baseline for object detection and tracking and illustrates LiDAR and image-based methods. In Caesar [28] a discussion of the role of pre-training and multiple LiDAR is given along with the detection performance. The paper mentioned how vital the matching function could be in impacting the ranking results of the detection methods. The experimental results of the detection and tracking works were reported to evaluate the performance of LiDAR-based versus image-based detectors. For the nuScenes dataset, it has already garnered much attention in the community of AVs and has been applied in various research.

Whereas in Ren and Yin [29] authors contributed a comprehensive review restricting 3D detection into the true scope of ADSs, which form a vital module for collision avoidance, motion prediction, and path planning. They identified some of the significant 3D object detection challenges, including recovering depth from images, learning from partial occlusions in point clouds, or aligning semantic information across modalities. In Ren and Yin [29] authors presented point cloud-based methods that mostly rely on LiDAR sensor data and exhibits higher accuracy and lower latency compared to image-based methods. To deal with the challenges of representation and processing of sparse, irregular, and unordered point clouds, they applied voxel-based and point-based methods. They also explored a point-voxel-based method that embodies both voxel and point-based methods. The research studied multimodal fusion-based approach to exploit the complementarity in different modalities like images and point clouds. Fusion methods with sequential and parallel approaches were considered in the work concerning sub-fashion sequential data flow and parallel data flow sets of the modalities as they pass through networks. On the one hand, the study indicated that point cloud-based methods are more accurate thanks to LiDAR sensor data. On the other hand, they are challenged by sparse, irregular, unordered point clouds in processing. In contrast, image-based methods operate based on visual information from cameras; they are cheap and interpretable but lack depth information. The paper enumerated the strengths and limitations of each technique but illustrated how they could work together effectively to attain 3D object detection for ADS.

Besides that, the work in Cui, et al. [30] presented an analysis of using DL techniques for camera-LiDAR fusion and image and point cloud data fusion in ADS. The authors provided a full review of techniques ranges from depth completion to object detection, semantic segmentation, and tracking. They tackled the trends in fusion methodologies: from 2D to 3D, from single-task to multi-tasks, and from signal-level to multi-level fusion. The research highlighted the current fusion pipeline needs to improve feature representation, integrate geometric constraints, and leverage temporal context. In the context of 2D/3D semantic segmentation, they described all the significant methods to fuse image and point cloud data under feature-level and result-level fusion. The feature-level fusion is when combining point cloud and picture features at the feature level. Techniques, such as multi-stage feature-level fusion and NASNet-based auto-encoder networks, are applied for 2D semantic segmentation. For 3D semantic segmentation, methods like 3DMV or UPF use both point cloud and multi-view image data and combine them after estimation of semantic labels per pixel or voxel. Result-level fusion involves data aggregation at the result-level and often helps leverage off-the-shelf 2D object detectors to narrow the 3D object detector's region of interest. Thus, it reduces computations by processing fewer regions of interest. Some examples are the frustum-based techniques such as F-Point Nets, Roar Nets, LiDAR Stereo Nets, etc. These algorithms project 2D bounding boxes into a 3D space for 3D object detection.

Finally, AVs can be a fascinating development, which could significantly reduce traffic accidents and fatalities; these are caused mainly by human error—speeding, intoxicated driving, and distracted driving [31]. At the very time, there exist many technological, legal, and ethical challenges that affect AVs and their comprehensive implementation.

5. Proposed System

The technology for ADS has been advanced because of the integration of ML techniques, with a special mention of CNN. This research implements edge detection and LKA using CNN with datasets [32]. Our goal is to improve perception and control systems in the AVs for safer and more efficient movement. This section focuses on the techniques and tools used in our work, such as MATLAB and Simulink.

5.1. Data Acquisition

Developing efficient CNNs for edge detection in AVs using structures starts with records accumulation that impacts the performance and generalization capabilities of the trained models. The essential factors of information gathering are described below:

- Kaggle dataset: Dataset from Kaggle available at "https://www.kaggle.com/datasets/alincijov/self-driving-cars" is used for this work due to its accessibility and comprehensiveness. This dataset contains labelled images that capture information about the driving environment.
- Image information: High-quality images are essential to capture details in the environment. The dataset we used provides a range of driving profiles, ensuring that the CNN models are trained on different datasets.
- Labelled data: The adopted dataset comes with descriptions that label features which is important for supervised learning. Labels include numbers from 1 to 5 that correspond to 'car', 'truck', 'pedestrian', 'bicyclist', and 'light' respectively.

5.2. Data Collection

The use of CNN for edge detection present several challenges, such as optimizing the training data. For the training to be effective, it is important to ensure a consistent and diverse set of data. To overcome this, we use rotation and inversion methods for data. One other major challenge is managing the quality of the training data. Ensuring that the dataset is diverse and accurately labelled is essential for effective training. We address this by using data augmentation techniques, such as flipping, rotating, and adding noise to the images, to create a more robust and varied training set. The main techniques used for data collection include the following:

- Sensor Integration: This relies on picture statistics collected from various riding scenarios. The sensors involved encompass cameras that retrieve information contained within the forms of photos. Large pixel applications are required for intending excellent edges as well as information within the environment.
- Manual Annotation: The edges and functions contained in the compound of pixels are labelled by human annotators. This procedure minimizes possible errors that may occur during extraction of features for side detection and is critical step in training of CNN models.
- Data Pre-processing: Pre-processing of the raw data involves cleaning, normalization and transformation to enhance its performance in the models in the train of CNN. This involves initial data cleaning, normalization, feature augmentation followed by dataset partitioning into training, validation and test sets.
- Data Cleaning: The process involves removing incorrect files, filling in incorrect labels, and ensuring all data points are correctly annotated. This step is essential to maintain the integrity of the dataset.
- Normalization: Normalizing the image data to a consistent range of pixels' values (typically between [0, 1] or [-1, 1]) helps accelerate the training resolution for CNN and improves model performance by ensuring uniform input to the networked CNN.

- Data Augmentation: This includes rotation, scaling, translation, and flipping. These are techniques in data pre-processing that help and improve the diversity of the training dataset in CNN trained model. Data augmentation improves model generalizability by exposing it to a broader range of environments.
- Edge Detection: It is an important step in self-driving cars in analysing the vehicle's environment since it identifies lane detection, road limits, and other vital elements. One of the techniques used is the Sobel operator for edge detection due to its efficiency and effectiveness.
- Sobel Operators: The Sobel operator computes the gradient of image intensity at each pixel, highlighting areas with high spatial frequency that correspond to edges. It employs two 3x3 convolution kernels to compute the gradient in the x and y directions. This is implemented using MATLAB by employing the Sobel filter to detect edges in grayscale images. The resulting gradient magnitude image detects the edges for the vehicle, and is utilized as input for additional processing in lane detection and LKA.

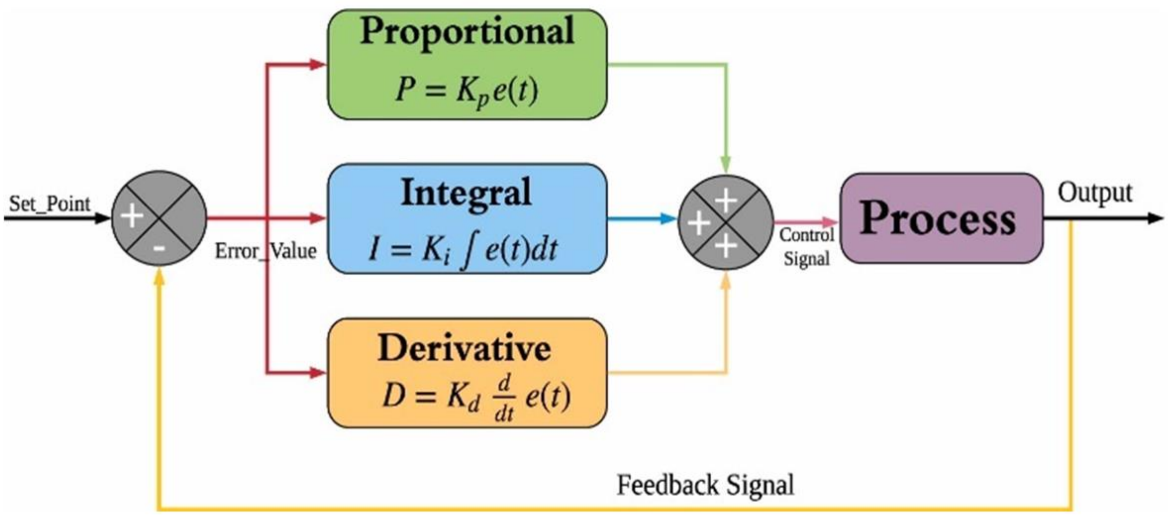


Figure 7.
PID controller architecture.

- Gradient Calculation: By convolving the image using Sobel kernels, the gradient in the x and y directions is determined to highlight the image's edges. Edge detection using the Sobel operator is essential for lane detection and object recognition as it gives a clear representation of the structural characteristics in the picture, allowing the autonomous system to accurately recognize lane boundaries and obstructions.
- Lane Detection: It uses the Hough Transform and edge detected in the road that AV moving on it to determine the actual lane markings. This step is critical for preserving the vehicle's place within the lane which assists the LKA feature.
- Hough Transform: The Hough Transform is used to identify straight lines in the edge detected for the road. It operates by mapping points in the image space to the Hough space and identifying lines based on parameter values even in the presence of noise. Peaks in the Hough space represent potential lines in the image. The Hough Transform is noise-resistant and can detect lines in a variety of different scenarios, making it excellent for identifying lane markers on road. This is critical for dependable lane keeping, especially in adverse conditions.

ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

5.3. PID Control

The PID control is a fundamental control technique that is used in a multiple and different technologies for ADS which include LKA in AVs. It is supposed to keep the car in its lane by constantly modifying the steering depending on feedback from the identified lane boundaries on the road. The PID controller, shown in Figure 7 Bhookya, et al. [33] calculates the steering angle by taking the angles into account, and the difference between the vehicle's present and desired positions (the lane centre) for the AV. This control mechanism enables smooth and responsive steering adjustments for vehicle, which improves the vehicle's ability to remain centred in its lane boundaries in the road. Below is an explanation of the role of PID control in LKA that is one of the most important technologies in ADS [34]:

5.4. Proportional (P) Control

- The proportional term generates an output according to the present error value. It computes an ideal steering angle based on the vehicle's deviation from the lane centre.
- The integral term is the accumulation of past errors.
- All the errors are summed up from the previous step over time which produces a corrective action that accounts for the cumulative error.
- This helps to eliminate steady-state errors that proportional controller alone cannot handle.

5.5. Derivative (D) Control

- The derivative term predicts future error from the steering based on its rate of change.
- It produces a corrective action proportional to the rate of change of the error, helping to damping oscillations and improving stability.

The total PID control output is a sum of proportional, integral, and derivative values. The PID controller calculates the steering angle by adding these three components.

5.5.1. Role of PID Control in LKA

- Error Calculation: The first step is to measure the error of line, which is the difference between the AV's current position and the centre of the lane. This error serves as the input to the PID controller.
- Proportional Action: The proportional term makes correction based on current errors that come from the first step. If the car is far from the lane centre of the line, the proportional action causes a higher corrective steering angle to bring the car back to the centre.
- Integral Action: The integral term accumulates error over time. If the AV has been repeatedly off-centre, the integral action changes the steering to fix the accumulated error and return the vehicle to the lane centre.
- Derivative Action: The derived term predicts the future trend of errors caused by the vehicle by measuring the rate of change. When the vehicle is accelerating off centre, the resultant action creates a damping effect, reducing corrective action to prevent overthrow and vibration.
- Control Output: The PID controller sums the outputs of the proportional, integral, and derivative actions to compute the final steering angle. This steering angle is then applied to the AV's steering system to adjust its direction between lanes to make LKA for ADS.

5.5.2. Importance of PID Control in LKA

The PID control provides smooth and rapid steering adjustments, which are critical for keeping the car in the lane. This ensures a safe and comfortable driving experience by reducing the likelihood of lane deviation and collisions. The PID controller continuously reacts to changes in the vehicle's position and the road environment to help the AV to be between the two lines of the LKA, allowing for accurate control of the vehicle's trajectory.

5.6. CNN for LKA

In the era of AI and DL, AVs have become a focal point of research and development. The primary goal of these efforts is to enable AVs to make decisions for hazard avoidance. One of the most critical approaches in training AVs is the use of DL and CNNs [35]. These techniques play a crucial role in ensuring that AVs can accurately stay in their lanes on public roads by fully perceiving the surrounding environment and identifying the lanes around the vehicle.

The CNN architecture use has a couple of convolutional layers, each is intended to extract numerous feature levels from the enter images. The series layers that come after these decrease the spatial dimensions of the information while preserving the necessary facts to make the community more computationally powerful. To guarantee robustness and generalizability, the community is trained on an extensive dataset of road images, encompassing a variety of driving conditions. Through this training technique, the community's weights are modified to reduce detection errors and enhance the precision of lane detection so facilitating decision-making in real-time based on sensory inputs. So, CNNs can give correct and timely information about the surroundings across the car.

The structure of our proposed CNN model consists of several convolutional layers, each followed by activation functions and pooling layers. The initial layers focused on identifying low-level features, such as beach textures, while the deeper layers had more complex shapes, including road markings, road boundaries and obstacles. The final CNN layers are designed accordingly to provide an accurate coastal map, which highlight critical areas of need for safe navigation.

Our proposed system utilizes a camera to capture the area around the vehicle and examines the perception plans. This approach is embodied in the LKA system. The primary function of LKA is to ensure that the vehicle maintains its lane. By leveraging advanced perception technologies, the vehicle can detect and follow lane markings, thus maintaining its path and avoiding unintended lane departures. As the depth of a CNN increases, elaborated in this structure, it gains the capacity of learning many features and patterns of different levels of hierarchy. In the first layers, the structure finds pragmatic primitive geometries within a picture including edges, possible simple textures and the like. As this data enters the deeper layers of the network, it starts to recognize increasingly higher levels of patterns such as shapes and forms of objects encountered. For instance, the initial layers might identify the edges of objects, while later layers might see further details such as the leaves that compose the trees or the individual elements of a car's tires. This hierarchical feature extraction makes CNNs well suited for application in autonomy of vehicles, especially AVs because a vehicle has to understand the environment as it tries to interpret what it sees. By capturing and processing such segregated information, CNNs enable featured recognition and perception of the environment by the AVs, proper identification of objects within this environment as well as balanced and informed driving decisions within this environment. This capability makes CNNs a preferred algorithm in the creation of AV inventions since most organizations would prefer to deal with a reliable and dependable algorithm [36].

5.7. CNN for Edge Detection

The CNN layers are built on the convolutional layer, which scans the input image and generates feature maps using convolutional filters (kernels) [37]. The community's capability to precisely perceive edges below a range of situations, which includes various lighting, climate, and sorts of roads, ensures that self-driving can depend upon correct records and selections. Below is an explanation of how this process works:

- Convolution Operation: A convolutional filter (a small weight matrix) is applied to the input image. At each point, the filter conducts element-wise multiplication and adds the results, producing a single value in the output feature map. This operation is repeated for full image.

ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

- Edge Detection Filters: Filters are intended to respond aggressively to regions with a high intensity contrast, such as edges. Examples of such filters include the Sobel Filter.
-

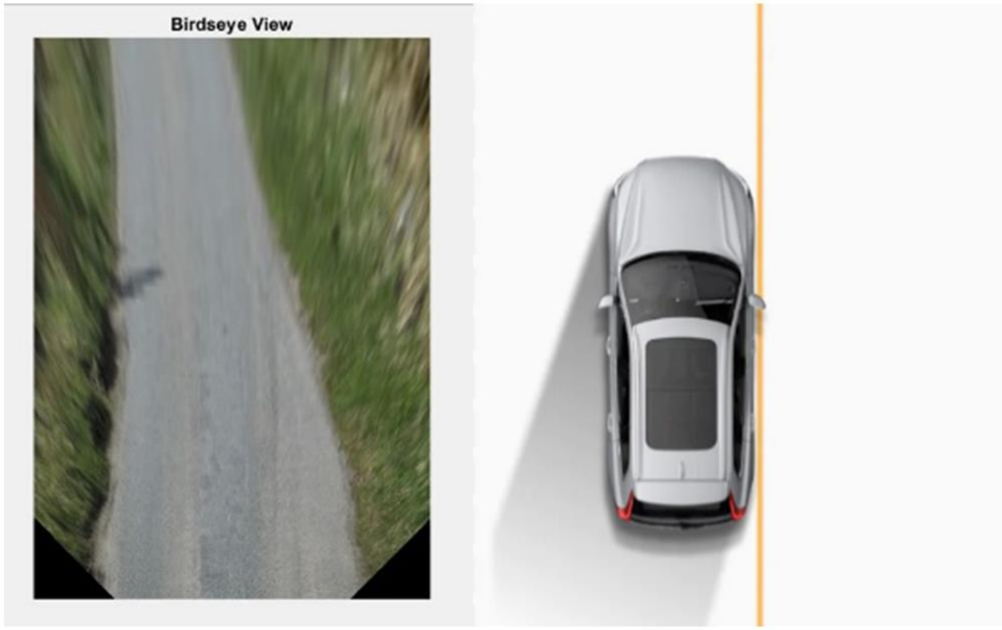


Figure 8.
Bird's eye view for ADS.

- Activation Functions: After the convolution process, the result is sent through an activation function, often known as the rectified linear unit, introducing nonlinearity into model.
- Activation: This converts all negative values in the feature map to zero, allowing the model to learn complicated patterns and features by stacking many layers.
- Pooling Layers: These minimize the spatial dimensions of feature maps while keeping the most critical data. This method is known as down sampling or subsampling.
- Max Pooling: It is the most frequent method that extracts the maximum value from each region of the feature map, highlighting the most important qualities (such as edges) while reducing computational complexity.
- Hierarchical Feature Learning: As the image moves through many convolutional and pooling layers, the CNN learns hierarchical characteristics.
- Early Layers: These layers usually detect low-level features like edges, lines, and corners. Intermediate layers integrate the low-level information to recognize more complex patterns and forms.
- Deeper Layers: They identify high-level, abstract features such as individual things or sections of objects.

6. Simulation and Results

Using modern hardware accelerators, inclusive of Graphics Processing Units (GPUs), which dramatically speed up the computation, will increase the CNN's capacity for real-time processing. As a result, the automobile can react fast to dynamic adjustments consisting of CNNs with additional computational and sensory. In our work, training deep CNNs required significant processing power and

ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

memory. To overcome this, we utilized cloud-based GPU resources, which provided the necessary computational capabilities.

6.1. Bird's Eye View

The bird's eye view shown in Figure 8, also known as top-down view, is an important perspective in ADS. It transforms road layout into a bird's eye view, and then detects the edges of the road within that transformed image using the Sobel edge detection method and Hough transform. We provide the bird's eye view code we used in Figure A1. The bird's-eye view is crucial for ADS applications where understanding the road layout from an overhead perspective is important for navigation and LKA. The importance of Bird's Eye View in ADS is due to the following reasons:

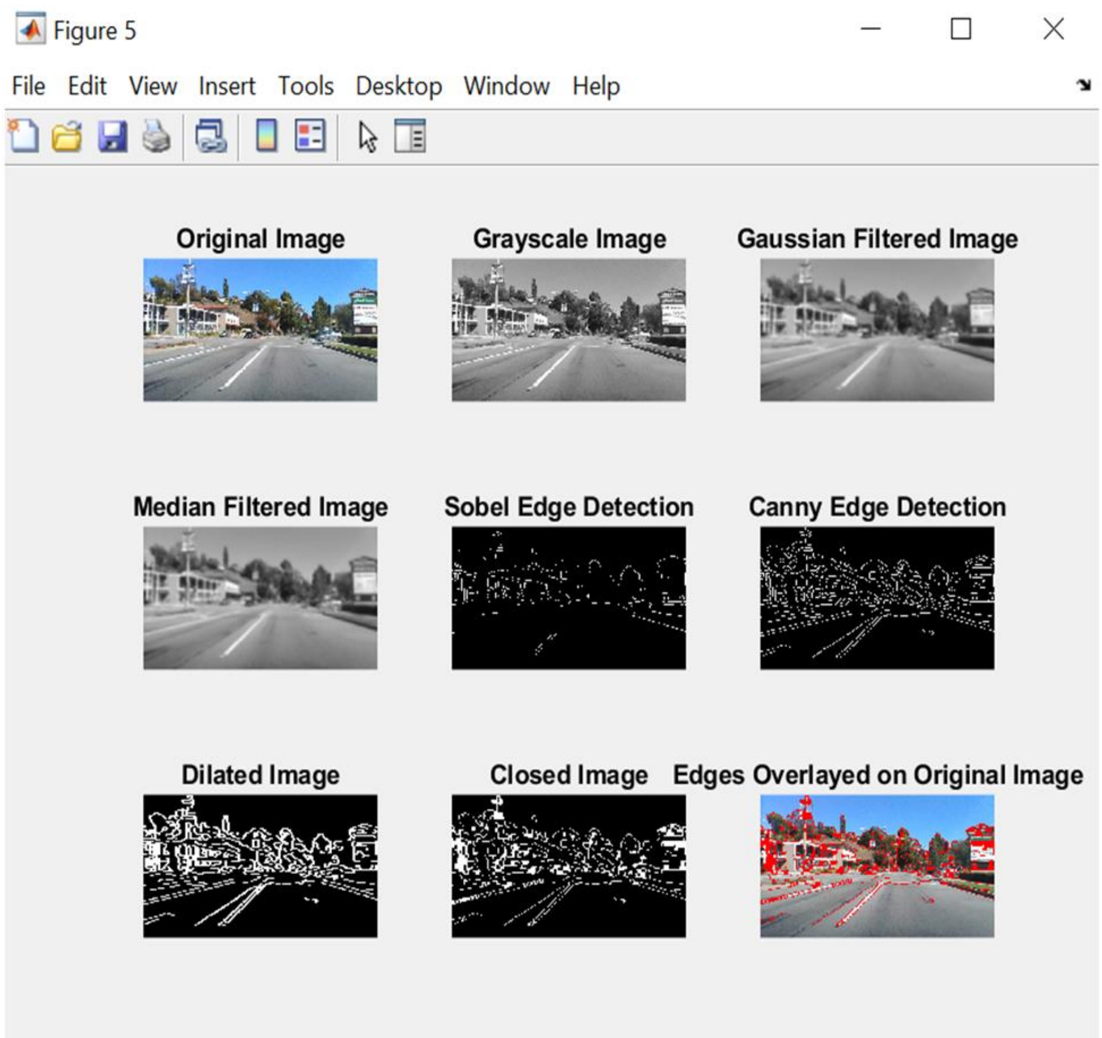


Figure 9.
Different types of filters.

- Enhanced Spatial Awareness: Offers a complete and clear view of the vehicle's surrounds, which is essential for navigation and avoiding obstructions.

DOI: 10.55214/25768484.v9i4.6259

- Accurate Lane Detection: Assists in accurately identifying lane lines and keeping the car in its lane.
- Improved Path Planning: Helps to design paths around obstacles with high precision, resulting in smoother and safer travel.
- Integration of CNNs: The bird's-eye view gives valuable data for CNN training, so enhancing its capacity to detect features and make top-down choices.

6.2. Dataset Filtering

The filtering for dataset or road makes various edge detection filters as shown in Figure 9, each of which contributes to the enhancement of image features in its own distinctive way. These improvements are crucial for CNNs because they establish an effective basis for hierarchical feature extraction and improve image processing accuracy. Integrating edge detection with CNNs improves training efficiency, feature relevance, and overall performance in real-world applications, making it critical pre-processing step in ADS and other Computer Vision tasks.

6.2.1. Applying Filters

The code in Figure A2 reads an image from dataset and displays it with the title "Original Image" to detect. Then, the image is converted to grayscale, which is a necessary step for many edge detection algorithms. After that, the following filters are applied:

- Sobel Filter: It is applied to detect edges by emphasizing regions of high spatial gradient.
- Prewitt Filter: It is another edge detection filter that highlights edge by detecting horizontal and vertical gradients.
- Roberts Filter: It is used for edge detection by calculating the gradient of the image intensity at each pixel.
- Canny Filter: It is a multi-stage edge detector that is popular due to its reliability and precision in detecting edges.

6.2.2. Importance of Filters in Edge Detection

- Sobel Filter: It detects changes in intensity, which makes it useful for emphasizing edges. The Sobel Filter exhibits directional sensitivity, that is, it is sensitive to both horizontal and vertical edges, yielding a comprehensive edge map.
- Prewitt Filter: It performs simple gradient calculation. The Prewitt filter is simple and effective for detecting edges, particularly in images with significant intensity shifts. Prewitt filters are computationally efficient, which makes them ideal for real-time applications.
- Roberts Filter: It has high precision. Roberts filters detect edges with great precision, particularly small details, making them ideal for detailed image analysis. They are more susceptible to noise, which can be reduced using pre-processing processes.
- Canny Filter: It provides multi-stage detection that incorporates noise reduction, gradient calculation, non-maximum suppression, side monitoring and hysteresis usage, resulting in remarkable aspect detection. Its strength is that it withstands noises and changing illumination situations, making it best for difficult picture evaluation jobs.

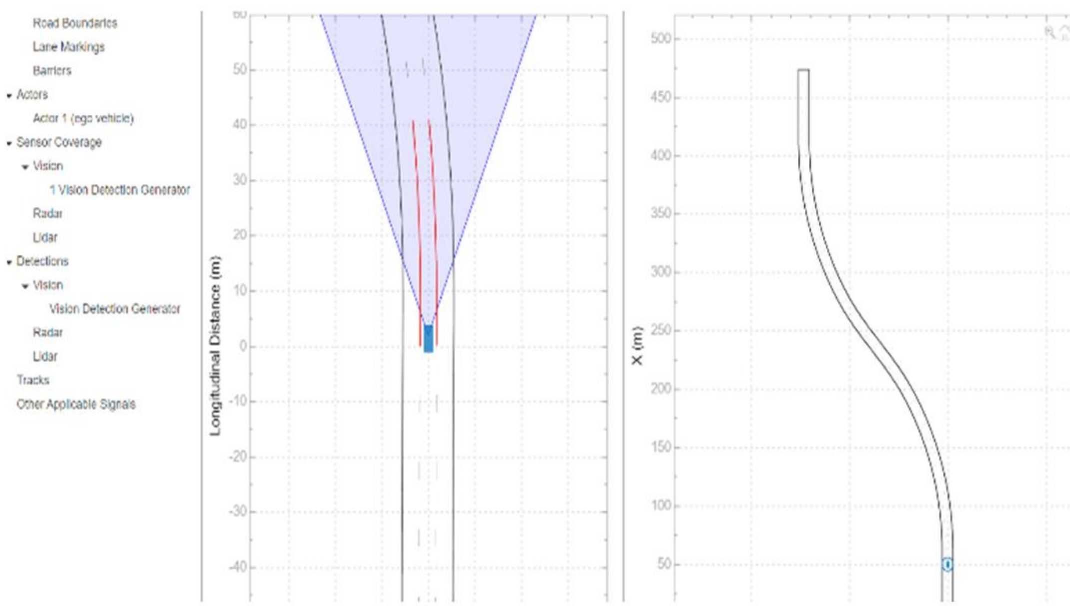
6.3. LKA Simulation with Control

The lane keeping code is intended to do lane detection and then help in LKA for AVs via edge detection and the Hough transform. This code uses an image to recognize lanes and calculate the necessary steering angle. The major control mechanism proposed for future iterations is varying the steering angle based on the vehicle's position relative to the detected lane. The LKA code outlines a method for detecting lane boundaries using edge detection and the Hough transform, which is crucial for developing an LKA system.

The AV detects the lines and edges and moves between the detected edges (line boundaries). The PID control is an essential method in ADS, including LKA, which involves continuously modifying the steering based on feedback from recognized lane boundaries. It determines the steering angle for the vehicle by taking into consideration the difference between the vehicle's current and ideal locations, ensuring that the car stays centred in its lane as demonstrated in Figure 10. This mechanism delivers smooth and rapid steering changes, which improves the vehicle's lane-keeping performance.

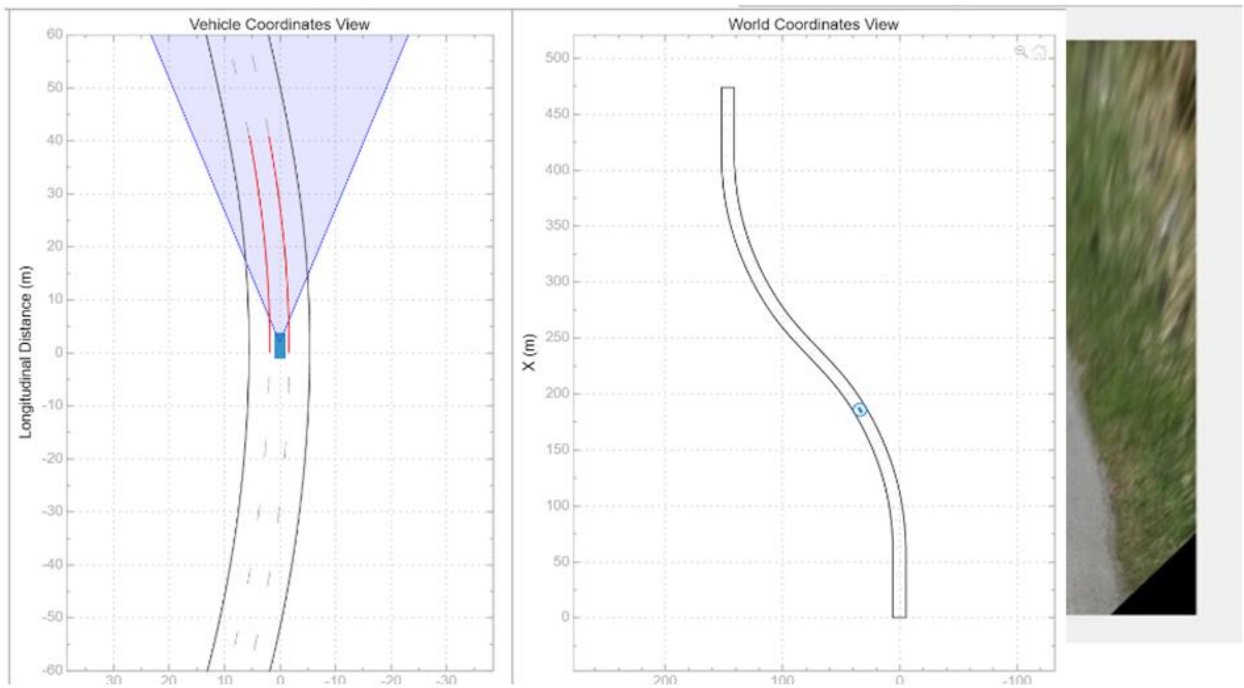
6.4. Lane Detection

The MATLAB code we used is intended to train CNN which requires lane recognition for AV systems. This method begins by defining the access methods to the dataset and model. Then, the code loads and pre-processes the dataset, splitting it into training and validation sets. The CNN is trained on a dataset of lane images, and its validation accuracy demonstrates the model's ability to accurately identify lane boundaries. Our results, given in Figure 11, show that the trained network can effectively detect lanes in a variety of driving scenarios, yielding an improved lane-keeping performance in AVs. Accuracy and processing time are the performance parameters we used to evaluate our CNN model [38]. The model's accuracy on the validation set shows its high accuracy of 100% as given in Figure 11. Finally, the trained model is successfully stored and prepared for deployment in lane detection applications.



(a)

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259



(b)

Figure 10 a, b.
LKA simulation with control.

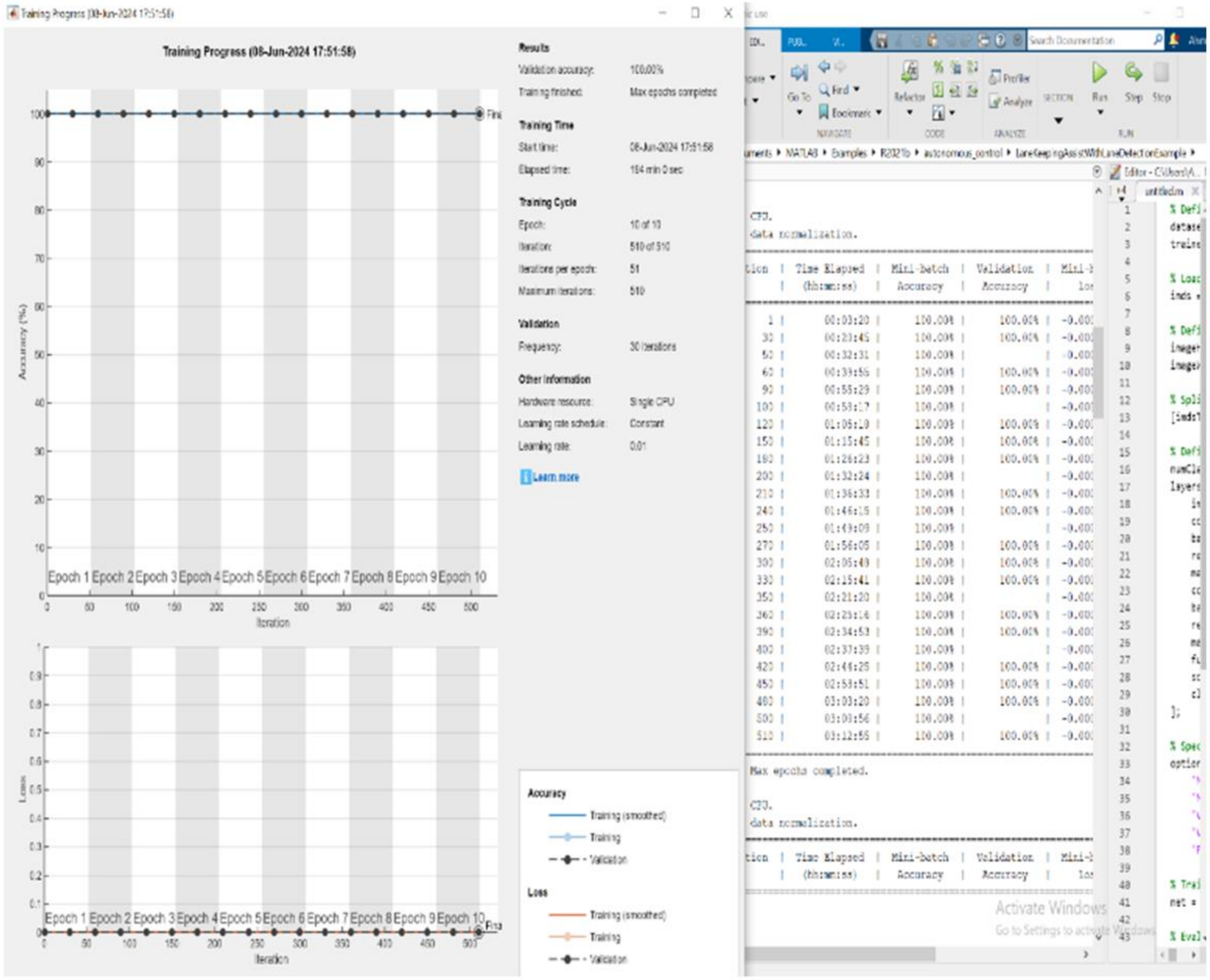


Figure 11.
CNN train model results.

6.5. Insights for Achieved Results

Autonomous car accidents have occurred, garnering a lot of attention from the media. Famous events include an Uber crash in Arizona in 2018 that claimed the life of a pedestrian, and a Tesla accident back in 2016 that killed the driver because the car failed to detect a truck. These incidents highlight the necessity of explicit guidelines for users of AVs as well as the enhancement of emergency protocols and sensory apparatus. Thus, the need for 100% accuracy is vital to avoid such accidents. Since our proposed system yields an accuracy of 100%, so it promotes the wide adoption of safe AVs in real-life environments.

7. Comparison with State-of-the-art Approaches

7.1. Comparison of CNN with PID Controller versus RNN with LSTM Cells for LKA

- CNN with PID Controller – Proposed Approach: CNNs are used to process and analyse visual features from cameras and sensors. They can recognize patterns and attributes, including lane markings, road signs, and obstacles. CNN analyses visual data to identify lane borders and other pertinent elements. Also, they can learn from big data and handle driving situations. While, PID

DOI: 10.55214/25768484.v9i4.6259

controllers exhibit low voltage, respond to immediate errors, and can struggle with dynamic and unpredictable systems.

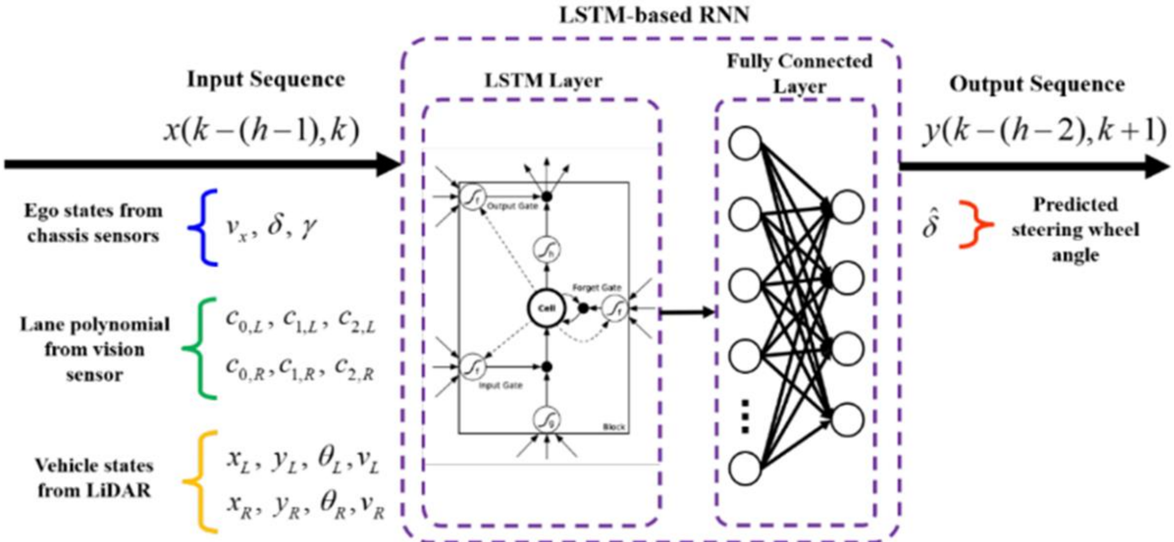


Figure 12.
LSTM-Based RNN for LKA.

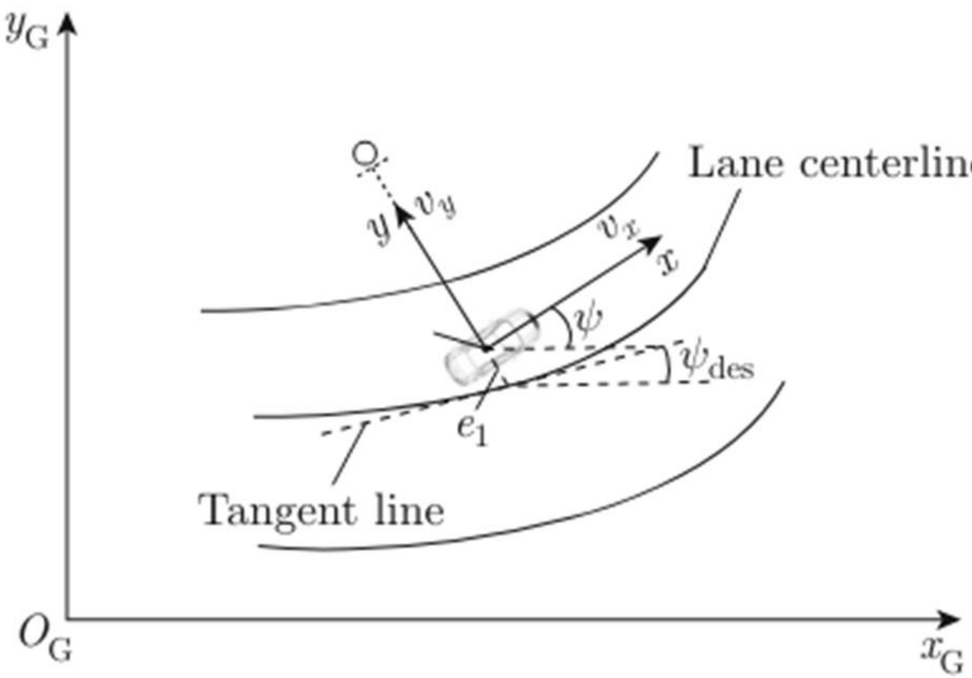


Figure 13.
MPC controller.

Also, they adjust the steering angle based on the CNN output, attempting to minimize the error between the current and intended positions to make vehicle in the centre between lanes. They adjust the

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259

steering wheel based on the difference between the current position of the vehicle and the centre of the desired path. PIDs use matching, coupling, and derivatives to make correct actions.

- RNN with LSTM Cells – Alternative Approach: RNN use LSTM cells to compute road markings and interactions with other vehicles. This method uses a sequence of inputs and previous states to accurately determine the inputs to the steering wheel as given in Figure 12 [39]. It uses memory capacity to account for past states and sequences of inputs, resulting in highly contextual predictions of steering object inputs. Also, it is highly adaptive since it takes into consideration temporal dependencies and input sequences, making it more capable of handling dynamic and unpredictable situations.

7.1.1. Implementation Complexity

- CNN with PID Controller: This offers moderate complexity. The CNN needs training, but the PID controller is simple to build and tune. Also, the CNN processes the images, and the PID controller instantly adjusts the steering wheel angle according to the results. Overall, good real-time performance.
- RNN using LSTM Cells: This offers high complexity. To address temporal dependence, significant training on data sequences is required, as well as more complex architecture. Real-time performance can be difficult to achieve due to the computational requirements of processing input sequences and predicting future states.

7.2. Comparison of Model Predictive Control versus PID Control for LKA

- Model Predictive Control (MPC): The adaptive MPC predicts future vehicle states and optimizes steering angle accordingly as illustrated in Figure 13 [40]. The MPC model can modify the steering depending on the road curvature prediction and present vehicle dynamics.
- PID Controllers: They cause changes in the steering angle according to the amount of deviation of the vehicle's position from the target position using the three main terms: proportional, integral, and derivative that are discussed earlier.

Comparison between MPC and PID Controllers is given in Table 1.

Table 1.
Comparison between MPC and PID Control for LKA.

Criteria	MPC	PID Control
Benefits	Improved lane keeping accuracy and responsiveness by taking into consideration the vehicle's future trajectory. Enhanced stability and control, particularly under dynamic conditions.	Easy to use and adjustable, gives sensitive and smooth control, and perfectly maintains the lane position in normal driving conditions.
Operation	Predictive, using a model of the vehicle's dynamics to forecast future states and optimize control inputs accordingly.	Based on error correction, using proportional, integral, and derivative terms to minimize the difference between the current and desired lane position.
Response	Proactive, considering future trajectories and adjusting control inputs accordingly.	Reactive, based on current and past errors.
Strength	Very flexible in a variety of dynamic and unpredictable settings.	Performs effectively in straightforward, predictable settings.
Limitation	Accurate forecasts entails a good model of the vehicle and surroundings.	Because of its reactive nature, its performance is affected by complicated and dynamic contexts.

8. Conclusion and Future Directions

Our paper showed the importance of combining the PID controllers with CNNs to improve LKA system in autonomous cars. It is clear that controlling the lane and reducing the chances of accidents due to wrong lane changes depends highly on accurate lane detection. CNNs are very good at processing visual input from cameras mounted on vehicles, so we can extract complex hierarchical features to recognize lane markers. This ensures LKA can detect lanes in all kind of driving scenarios, including low light and bad weather. This makes ADSs more reliable and safe, so our work helps to advance ADSs through the usage of most advanced technologies, leading to a better future transportation.

PID controllers are the control part of LKA systems, they enhance CNNs perceptual ability by providing strong steering correction based on real-time feedback from the detected lane boundaries. The vehicle position in the lane is maintained by this control system to provide responsive and smooth steering correction so the car can be centred. PID controllers are good for LKA systems because they are simple and efficient. They can adjust the steering angle to correct any error from the original direction, so that driving can be safe and beneficial.

The combination of CNN with PID controller provides a robust LKA system with full potential. On one side, CNNs supply the perceptual competencies for line detection; on the other side, PID controllers ensure precision and responsiveness in steering adjustment. In this manner, LKA systems are able to maintain rigorous discipline within lanes against the unpredictable dynamic conditions of driving. Besides improving LKA systems' performance, CNN-based perception and PID-based control improve the overall safety and reliability of ADS technologies. Our simulation and results obtained point out that our CNN-based approach yields high accuracy and low processing time that are needed for real-time situations.

Several critical paths will dominate future research in this area, all of which can improve the potential of LKA systems for real-world deployment. Future work could go in the following directions that span around improving performance and security of model:

- More Complex CNN Architectures: This is needed to examine advanced CNN architectures and methods to improve the robustness of lane detection and other perceptual tasks.
- Hybrid Models: CNNs could be combined with other ML models, such as LSTM networks or RNNs, to capture temporal relationships in driving data. This would visibly improve the system adaptability.
- Additional Sensors: Multi-modal sensor fusion could be used to utilize information from other sensors, such LiDAR, radar, and ultrasonic sensors, in order to improve dependability of the perception system, especially in difficult situations like dimly lit areas or unfavourable weather.
- Sensor Calibration and Synchronization: In order to ensure accurate data fusion and enhance overall system performance, techniques for the exact calibration and synchronization of many sensors could be developed.
- Optimized Algorithms: This is required to ensure that the system can run effectively on embedded hardware with limited resources, so focusing on optimizing the models and algorithms would lower computational complexity and increase real-time performance.
- Hardware Acceleration: Examining how processing activities, involving perception and control, can be accelerated by using hardware accelerators.
- Robustness in Response to Environmental Changes: Studying domain adaptation strategies that improve the system's resistance to changes in the weather, lighting, and road conditions is crucial to develop a more effective model that generalizes to a variety of situations.
- Driver Monitoring Systems for Human-machine Interaction: These could be integrated to enable a safe transition between autonomous and manual driving modes, especially when the autonomous system is having problems.
- User Interface Design: To ensure a smooth connection between the driver and the autonomous system, the user interface design needs to be robust to give the driver clear and intuitive feedback.
- Adversary Training: Use these techniques to strengthen the system's defences against possible adversarial attacks and guarantee dependable performance in real-life situations.

- Regulatory Aspects: In order to facilitate the deployment of the created ADS in real-world scenarios, work with regulatory bodies to ensure that it satisfies with current laws and regulations.
- Ethical Frameworks: Creating and putting into effect strong ethical frameworks and rules to address ethical issues related to ADS is vital, including areas as data privacy and decision-making in dangerous situations.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Acknowledgements:

The authors would like to thank all reviewers of this paper for their efforts and dedication in reviewing this work. All their valuable comments are added value to our work.

References

[1] M. B. M. Mansour and A. Said, "Multimodal routing for connecting people and events in sustainable intelligent systems." Singapore: Springer Singapore, 2021, pp. 267-282.

[2] M. B. M. Mansour, A. Said, N. E. Ahmed, and S. Sallam, "Autonomous parallel car parking," presented at the IEEE Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), pp. 392-397, 2020.

[3] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443-58469, 2020.

[4] F. Han, D. Li, and Q. Hao, "Autonomous driving framework for bus transit systems towards operation safety and robustness," presented at the In IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2778-2784, 2019.

[5] K. Geng and S. Liu, "Robust path tracking control for autonomous vehicle based on a novel fault tolerant adaptive model predictive control algorithm," *Applied Sciences*, vol. 10, no. 18, p. 6249, 2020.

[6] L. Liu *et al.*, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6469-6486, 2020.

[7] M. B. M. Mansour, "SCPP: Secure credential provisioning protocol for cellular vehicles," presented at the IEEE International Symposium on Systems Engineering (ISSE), pp. 1-8, 2019.

[8] M. B. Mansour, C. Salama, H. K. Mohamed, and S. A. Hammad, "CARCLOUD: A secure architecture for vehicular cloud computing," presented at the 14th Embedded Security in Cars Europe Conference, ESCAR, Germany, 2016.

[9] A. Dosovitskiy, "CARLA: An open urban driving simulator," in *Proc. of the 1st Annual Conference on Robot Learning (CoRL)*, 2017.

[10] C. Luetge, "The German ethics code for automated and connected driving," *Philosophy & Technology*, vol. 30, pp. 547-558, 2017.

[11] F. Golbabaei, T. Yigitcanlar, and J. Bunker, "The role of shared autonomous vehicle systems in delivering smart urban mobility: A systematic review of the literature," *International Journal of Sustainable Transportation*, vol. 15, no. 10, pp. 731-748, 2021. https://doi.org/10.1080/15568318.2020.1798571

[12] J. Guo, J. Wang, Y. Luo, and K. Li, "Robust lateral control of autonomous four-wheel independent drive electric vehicles considering the roll effects and actuator faults," *Mechanical Systems and Signal Processing*, vol. 143, p. 106773, 2020.

[13] M. Ryan, "The future of transportation: ethical, legal, social and economic impacts of self-driving vehicles in the year 2025," *Science and engineering ethics*, vol. 26, no. 3, pp. 1185-1208, 2020.

[14] D. Gruyer, V. Magnier, K. Hamdi, L. Claussmann, O. Orfila, and A. Rakotonirainy, "Perception, information processing and modeling: Critical stages for autonomous driving applications," *Annual Reviews in Control*, vol. 44, pp. 323-341, 2017. https://doi.org/10.1016/j.arcontrol.2017.09.012

[15] M. Heimberger, J. Horgan, C. Hughes, J. McDonald, and S. Yogamani, "Computer vision in automated parking systems: Design, implementation and challenges," *Image and Vision Computing*, vol. 68, pp. 88-101, 2017.

[16] D. Bhatt *et al.*, "CNN variants for computer vision: History, architecture, application, challenges and future scope," *Electronics*, vol. 10, no. 20, p. 2470, 2021.

[17] J. Li, W. Zhan, Y. Hu, and M. Tomizuka, "Generic tracking and probabilistic prediction framework and its application in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3634-3649, 2019.

[18] M. Geisslinger, F. Poszler, and M. Lienkamp, "An ethical trajectory planning algorithm for autonomous vehicles," *Nature Machine Intelligence*, vol. 5, no. 2, pp. 137-144, 2023.

[19] M. R. Bachute and J. M. Subhedar, "Autonomous driving architectures: insights of machine learning and deep learning algorithms," *Machine Learning with Applications*, vol. 6, p. 100164, 2021.

[20] A. Ammar, A. Koubaa, M. Ahmed, A. Saad, and B. Benjdira, "Aerial images processing for car detection using convolutional neural networks: Comparison between faster r-cnn and yolov3," *arXiv preprint arXiv:1910.07234*, 2019.

[21] S. Liang *et al.*, "Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25345-25360, 2022.

[22] Y. Zhang, A. Carballo, H. Yang, and K. Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 196, pp. 146-177, 2023.

[23] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315-329, 2020.

[24] P. Wang, S. Gao, L. Li, B. Sun, and S. Cheng, "Obstacle avoidance path planning design for autonomous driving vehicles based on an improved artificial potential field algorithm," *Energies*, vol. 12, no. 12, p. 2342, 2019. https://doi.org/10.3390/en12122342

[25] S. Rani, D. Ghai, and S. Kumar, "Object detection and recognition using contour based edge detection and fast R-CNN," *Multimedia Tools and Applications*, vol. 81, no. 29, pp. 42183-42207, 2022.

[26] J. Mao, S. Shi, X. Wang, and H. Li, "3D object detection for autonomous driving: A comprehensive survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 1909-1963, 2023.

[27] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multi-modal sensor fusion for auto driving perception: A survey," *arXiv preprint arXiv:2202.02703*, 2022.

[28] H. Caesar, "NuScenes: A multimodal dataset for autonomous driving," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020,pp.11621-11631*, 2020.

[29] B. Ren and J. Yin, "Sdvrf: Sparse-to-dense voxel region fusion for multi-modal 3d object detection," presented at the In 2024 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1856-1861). IEEE, 2024.

[30] Y. Cui *et al.*, "Deep learning for image and point cloud fusion in autonomous driving: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 722-739, 2021.

[31] J. Wang, L. Zhang, Y. Huang, and J. Zhao, "Safety of autonomous vehicles," *Journal of advanced transportation*, vol. 2020, no. 1, p. 8867757, 2020.

[32] I.-C. Sang and W. R. Norris, "Improved generalizability of CNN based lane detection in challenging weather using adaptive preprocessing parameter tuning," *Expert Systems with Applications*, vol. 275, p. 127055, 2025.

[33] J. Bhookya, M. V. Kumar, J. R. Kumar, and A. S. Rao, "Implementation of PID controller for liquid level system using mGWO and integration of IoT application," *Journal of Industrial Information Integration*, vol. 28, p. 100368, 2022.

[34] S. Wei, P. E. Pfeffer, and J. Edelmann, "State of the art: Ongoing research in assessment methods for lane keeping assistance systems," *IEEE Transactions on Intelligent Vehicles*, 2023.

[35] F. E. Morooka, A. M. Junior, T. F. Sigahi, J. d. S. Pinto, I. S. Rampasso, and R. Anholon, "Deep learning and autonomous vehicles: Strategic themes, applications, and research agenda using SciMAT and content-centric analysis, a systematic review," *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 763-781, 2023.

[36] N. U. A. Tahir, Z. Zhang, M. Asim, J. Chen, and M. ELAffendi, "Object detection in autonomous vehicles under adverse weather: A review of traditional and deep learning approaches," *Algorithms*, vol. 17, no. 3, p. 103, 2024.

[37] M. R. Islam, T. A. Siddique, M. I. H. Sakib, and S. Hossain, "A convolutional neural network for end to end structural prediction and lane detection for autonomous vehicle," presented at the In 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT) (pp. 1-6), 2021.

[38] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362-386, 2020. https://doi.org/10.1002/rob.21918

[39] Y. Jeong, "Interactive lane keeping system for autonomous vehicles using LSTM-RNN considering driving environments," *Sensors*, vol. 22, no. 24, p. 9889, 2022. https://doi.org/10.3390/s22249889

[40] W. Cao, C. Liu, Z. Liu, and Z. Wei, "Integrated LKA and DYC control of four-wheel-independent-drive electric vehicles with a central-zonal electronic and electrical architecture," *IEEE Transactions on Vehicular Technology*, 2024.

Appendices

Appendix A

Editor - C:\Users\Apple\Desktop\Mansorzz RP final 179019\Mansorzz RP\Mansorzz RP\birdseye.m

birdseye.m | filtering.m | helperLKASetUp.m | lane_keeping.m | untitled.m

```
%-----------------------------------------------------------------------
%           Convert Image to Birdseye view
%
% This was done using the documentation on the transformImage function
% available at https://uk.mathworks.com/help/driving/ref/birdseyeview.transformimage.html#bvh354t-1-birdsEye
%-----------------------------------------------------------------------
focalLength    = [309.4362, 344.2161]; %Define camera intrinsics
principalPoint = [318.9034, 257.5352];
imageSize      = [480, 640];

camIntrinsics = cameraIntrinsics(focalLength, principalPoint, imageSize);

%Set the area in front of the camera too be converted into a birdseye view
%i.e. 3 to 20m with 6m on either side of the camera
distAhead = 20;
spaceToOneSide = 6;
bottomOffset = 3;
outView = [bottomOffset,distAhead,-spaceToOneSide,spaceToOneSide];

height = 2.1798;    % height of camera from the ground (m)
pitch  = 14;        % pitch of the camera in degrees
sensor = monoCamera(camIntrinsics,height,'Pitch',pitch); %set camera configuration

imageSize = [NaN, 400]; % width = 250 pixels, NaN allows for automatic calculation of height to maintain aspect ratic
birdsEye = birdsEyeView(sensor,outView,imageSize); %initialise birdsEyeView object

BEV = transformImage(birdsEye,I); %Transform image to birdseye view

%-----------------------------------------------------------------------
%      Detect edges of the road and display on birdseye view image
%-----------------------------------------------------------------------

BEVG = rgb2gray(BEV); %Convert to black and white image

BW = edge(BEVG,'sobel'); %Detect edges in image

[H,T,R] = hough(BW, 'RhoResolution',0.5,'Theta',-88:88);
P  = houghpeaks(H,2,'threshold',ceil(0.3*max(H(:))),'NHoodSize',[95 95]);

lines = houghlines(BW,T,R,P,'FillGap',40,'MinLength',40);
figure, imshow(BEV), hold on
[rows, columns]=size(I);
max_len = 0;
```


Figure A1.
Bird's eye view code.

```
I = imread(imagePath);

% Convert the image to grayscale
I_gray = rgb2gray(I);

% Apply Gaussian filter to reduce noise
I_gaussian = imgaussfilt(I_gray, 2);

% Apply Median filter to further reduce noise
I_median = medfilt2(I_gaussian, [3 3]);

% Apply Sobel filter to detect edges
I_sobel = edge(I_median, 'sobel');

% Apply Canny edge detection for more refined edges
I_canny = edge(I_median, 'canny');

% Apply morphological operations to clean up the image
se = strel('line', 5, 0);
I_dilated = imdilate(I_canny, se);
I_eroded = imerode(I_dilated, se);

% Apply a closing operation to fill gaps in the detected edges
se = strel('line', 5, 90);
I_closed = imclose(I_eroded, se);

% Overlay the detected edges on the original image
I_overlay = imoverlay(I, I_closed, [1 0 0]); % Red color overlay

% Display the results
figure;
subplot(3, 3, 1), imshow(I), title('Original Image');
subplot(3, 3, 2), imshow(I_gray), title('Grayscale Image');
subplot(3, 3, 3), imshow(I_gaussian), title('Gaussian Filtered Image');
subplot(3, 3, 4), imshow(I_median), title('Median Filtered Image');
subplot(3, 3, 5), imshow(I_sobel), title('Sobel Edge Detection');
subplot(3, 3, 6), imshow(I_canny), title('Canny Edge Detection');
subplot(3, 3, 7), imshow(I_dilated), title('Dilated Image');
subplot(3, 3, 8), imshow(I_closed), title('Closed Image');
subplot(3, 3, 9), imshow(I_overlay), title('Edges Overlayed on Original Image');
```

Figure A2.
Code for filters using in ADS.

ISSN: 2576-8484
Vol. 9, No. 4: 1300-1331, 2025
DOI: 10.55214/25768484.v9i4.6259