A

Minor Project Report

on

Autonomus Vehicle Simulation

Submitted to

BHILAI INSTITUTE OF TECHNOLOGY, DURG

An Autonomous Institute

Affiliated to

CHHATTISGARH SWAMI VIVEKANAND TECHNICAL UNIVERSITY BHILAI(C.G.)

In partial fulfillment of the requirement for the award of

Bachelor of Technology

in

Computer Science and Engineering

By

Sumeet Singh	:- 300102222026
Aman Dewangan	:- 300102223307
Abhijeet	:- 300102222031

Under the Guidance of

Dr. Ashok Kumar Behra	Dr. Shankha De
Professor	**Professor**

Session 2024-2025

DECLARATION BY THE CANDIDATE

We, the undersigned solemnly declare that the report of the project work entitled "**Autonomus Vehicle Simulation**" is based on my own work carried out during the course of my study under the guidance of **Dr. Ashok Kumar Behra** and **Dr. Shankha De**.

I confirm that this work has not been submitted to any other institution for the award of any degree or diploma.

Date: __________
Place: __________

SIGNATURE OF STUDENTS :-

Sumeet Singh __________________

Aman Dewangan __________________

Abhijeet __________________

CERTIFICATE

This is to Certify that the report of the project submitted is an outcome of the project work entitled "**Autonomus Vehicle Simulation"** carried out by **Sumeet Singh**, **Aman Dewangan** and **abhijeet**.

Under my guidance and supervision in partial fulfillment of Bachelor of Technology in Computer Science from Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G).

To the best of my knowledge and belief the project

i) Embodies the work of the candidate himself / herself,

ii) Has duly been completed,

iii) Fulfills the requirement of Ordinance relating to the B. Tech. degree of University

iv) Is up to the desired standard for the purpose of which is submitted.

Signature of the guide

Dr. Ashok Kumar Behra   Dr. Shankha De

________________   ________________

The Project work as mentioned above is hereby being recommended and forwarded for examination and evaluation.

Dr. (Mrs,) Sunita Soni

HOD(COMPUTER SCI AND ENGINEERING)

CERTIFICATE BY THE EXAMINERS

This is to Certify that the project the entitled

PROJECT TITLE : "Autonomus Vehicle Simulation",

Submitted by

Name : Sumeet Singh, ENROLLMENT NO. CC-1524

Aman Dewangan, ENROLLMENT NO. BJ-7500

Abhijeet , ENROLLMENT NO. CC-1526

Have been examined by the undersigned as a part of the examination for the award of Bachelor of Technology degree in Computer Science and Engineering from Bhilai Institute of Technology, Durg, an autonomous institute affiliated to Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G)

(Internal Examiner)	**(External Examiner)**
Name:	**Name:**
Date:	**Date:**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Dr. Ashok Kumar Behra** and **Dr. Shankha De**.

my project guide, for their invaluable guidance, encouragement, and support throughout this project. Their expertise and feedback were instrumental in shaping the development of the “**Autonomus Vehicle Simulation”**.

We would also like to express my/our gratitude towards our Head of Department Dr. Mrs. Sunita Soni for giving me/us this great opportunity to do a project on “**Autonomus Vehicle Simulation”**.

Without their support and suggestions, this project would not have been completed.

(Signature of the candidate)

Name: Sumeet Singh

(Signature of the candidate)

Name: Aman Dewangan

(Signature of the candidate)

Name: Abhijeet

List of Figures

CONTENTS

Chapter 1: Introduction

The field of autonomous driving has seen remarkable advancements in recent years, driven by the integration of artificial intelligence, deep learning, and computer vision. The primary challenge in autonomous vehicle development lies in enabling vehicles to understand and react to their surroundings in real time. This project, titled **"Automated Vehicle Simulation with Lane and Vehicle Detection,"** focuses on simulating the perception and decision-making process of a self-driving car using real-time image and video processing techniques. The project leverages the power of deep learning and classical computer vision to detect road lanes and recognize vehicles from camera input, creating an intelligent environment where the car can make informed decisions.

To detect vehicles, the system incorporates the YOLOv8 (You Only Look Once, version 8) object detection algorithm, a state-of-the-art model known for its high speed and accuracy in real-time environments. YOLOv8 enables the system to detect vehicles such as cars, motorcycles, buses, and trucks within each frame of video or image input. On the other hand, lane detection is achieved using classical computer vision methods such as grayscale conversion, Gaussian blurring, Canny edge detection, and Hough Line Transform. These techniques are used together to identify and draw road lane boundaries, simulating how autonomous vehicles maintain lane discipline.

The entire simulation is built in **Python**, utilizing essential libraries such as **OpenCV** for image processing, **NumPy** for array manipulation, and **Ultralytics' YOLO** for object detection. Additionally, the system is enhanced with a **Streamlit web interface**, which provides a simple yet interactive way for users to upload road images/videos or preview a live camera feed to observe detection results. The project serves as a fundamental prototype that demonstrates how machine learning and image processing can work together to emulate autonomous vehicle behavior, opening doors for further advancements in automated driving simulations and real-world applications.

Key Points

1. **Project Title**: Automated Vehicle Simulation with Lane and Vehicle Detection
2. **Objective**: To design a system that simulates self-driving car functionalities by detecting road lanes and vehicles using computer vision techniques.
3. **Technologies Used**:
    a. **YOLOv8** – for real-time object (vehicle) detection
    b. **OpenCV** – for image processing and lane detection
    c. **NumPy** – for numerical computations
    d. **Streamlit** – to develop an interactive web interface
4. **Input Sources**:
    a. Live webcam feed
    b. Uploaded images or video files of roads
5. **Vehicle Detection**:
    a. Identifies vehicles such as cars, motorcycles, buses, and trucks
    b. Draws bounding boxes and labels them appropriately
6. **Lane Detection**:
    a. Uses edge detection and Hough Line Transform to detect lane markings

    b. Highlights lane lines on the road surface in the video or image

7. **User Interface**:
    a. Provides an easy-to-use web application for file upload or live camera preview
8. **Expected Output**:
    a. Annotated visuals showing detected lanes and vehicles
9. **Applications**:
    a. Educational purposes, traffic monitoring, simulation of autonomous driving systems
10. **Scope for Future Work**:
    a. Can be enhanced to detect traffic signs, pedestrians, and dynamic decision-making for real-world self-driving simulations

Chapter 2: Problem Statement and Software Requirement

Problem Statement

In today's rapidly advancing world of technology and transportation, the need for intelligent systems capable of assisting or simulating autonomous driving has become increasingly critical. Despite significant advancements in vehicle automation, there remains a gap in affordable and accessible simulation tools that can effectively demonstrate core features of autonomous vehicles such as lane detection and vehicle recognition.

Manual driving is prone to human errors, especially in scenarios involving poor visibility, traffic congestion, and high-speed decision-making. Furthermore, the development and testing of self-driving systems require complex setups and real-world trials, which can be both costly and time-consuming. This project addresses the problem by creating an automated vehicle simulation that integrates real-time lane and vehicle detection using advanced computer vision techniques.

The system aims to provide a functional model that can identify road lanes and nearby vehicles from either a live camera feed or uploaded media, simulating how a self-driving vehicle perceives its surroundings. The goal is to design a web-based, interactive, and cost-effective prototype that can aid in education, research, and preliminary testing for autonomous systems.

Software Requirements

1. Programming Language

- **Python 3.10 or higher**
  Used for backend development, computer vision tasks, and running machine learning models.

2. Libraries and Frameworks

- **OpenCV**
  For image processing, edge detection, and camera handling.
- **NumPy**
  For numerical computations and handling image arrays.
- **Ultralytics YOLOv8**
  For real-time object (vehicle) detection.
- **Torch (PyTorch)**
  Required to run the YOLOv8 deep learning model.
- **Streamlit**
  Used for building the web-based user interface with video upload and live preview.

3. Development Environment

- **Jupyter Notebook / VS Code / PyCharm**
  For writing and testing the Python code.

4. Web Browser

- **Any modern browser (e.g., Chrome, Edge, Firefox) to view and interact with the Streamlit interface.**

5. Operating System

- **Windows 10 or higher**
  (The code can also run on Linux or macOS with slight adjustments.)

6. Others (Optional)

- **Git**
  For version control and collaboration.

- **Virtual Environment**
  For managing dependencies and isolating the project environment.

Chapter 3: Methodology

The methodology for this project involves a series of systematic steps aimed at developing an automated vehicle simulation system. The system detects lanes and vehicles using computer vision techniques and deep learning models to drive the vehicle autonomously. Below are the steps followed:

1. Data Collection

- **Image/Video Data**: The project uses camera input (from a webcam or uploaded video/image) to detect lanes and vehicles in real-time.

- **Pre-trained YOLOv8 Model**: A pre-trained model is utilized for vehicle detection using the **COCO dataset**.

2. Image Preprocessing

- **Convert to Grayscale**: Each frame is converted to grayscale to reduce computational complexity.

- **Blurring**: Gaussian blur is applied to smooth the image, reducing noise and making edge detection more efficient.

- **Edge Detection (Canny)**: Canny edge detection is used to identify significant edges in the image, particularly for lane lines.

3. Region of Interest (ROI) Masking

- **Defining Area of Interest**: A mask is created for the region of interest (ROI) in the image to focus on the road area, excluding unnecessary background and non-road features. This allows more precise lane detection.

4. Lane Detection

- **Hough Line Transform**: After detecting edges, the Hough Line Transform algorithm identifies and draws the lane lines (such as dashed or continuous lines) on the road.

5. Vehicle Detection

- **YOLOv8 Inference**: YOLO (You Only Look Once) is used for real-time vehicle detection. This object detection model identifies vehicles (car, bus, truck, motorcycle) in the frame by using bounding boxes around detected vehicles.

- **Class Identification**: The class IDs (from the COCO dataset) are used to filter out irrelevant objects and detect specific vehicles such as cars and trucks.

6. Combining Results

- **Lane Detection and Vehicle Detection**: After detecting lanes and vehicles separately, both are displayed on the same frame. Lane lines are drawn in green, and vehicles are marked with red bounding boxes.

- **Real-Time Processing**: Both lane detection and vehicle detection are performed in real-time, enabling the system to simulate an autonomous vehicle scenario.

7. Interface (Streamlit)

- **Web Interface**: A Streamlit-based interface allows users to upload images or videos for lane and vehicle detection.

- **Live Preview**: The live webcam feed is processed, and detected lanes and vehicles are displayed in real-time.

8. Testing and Evaluation

- **Validation**: The system is tested using various road scenarios to ensure correct lane and vehicle detection.

- **Performance Metrics**: Speed, accuracy of lane detection, and vehicle classification are evaluated.

DATA FLOW DIAGRAM:-

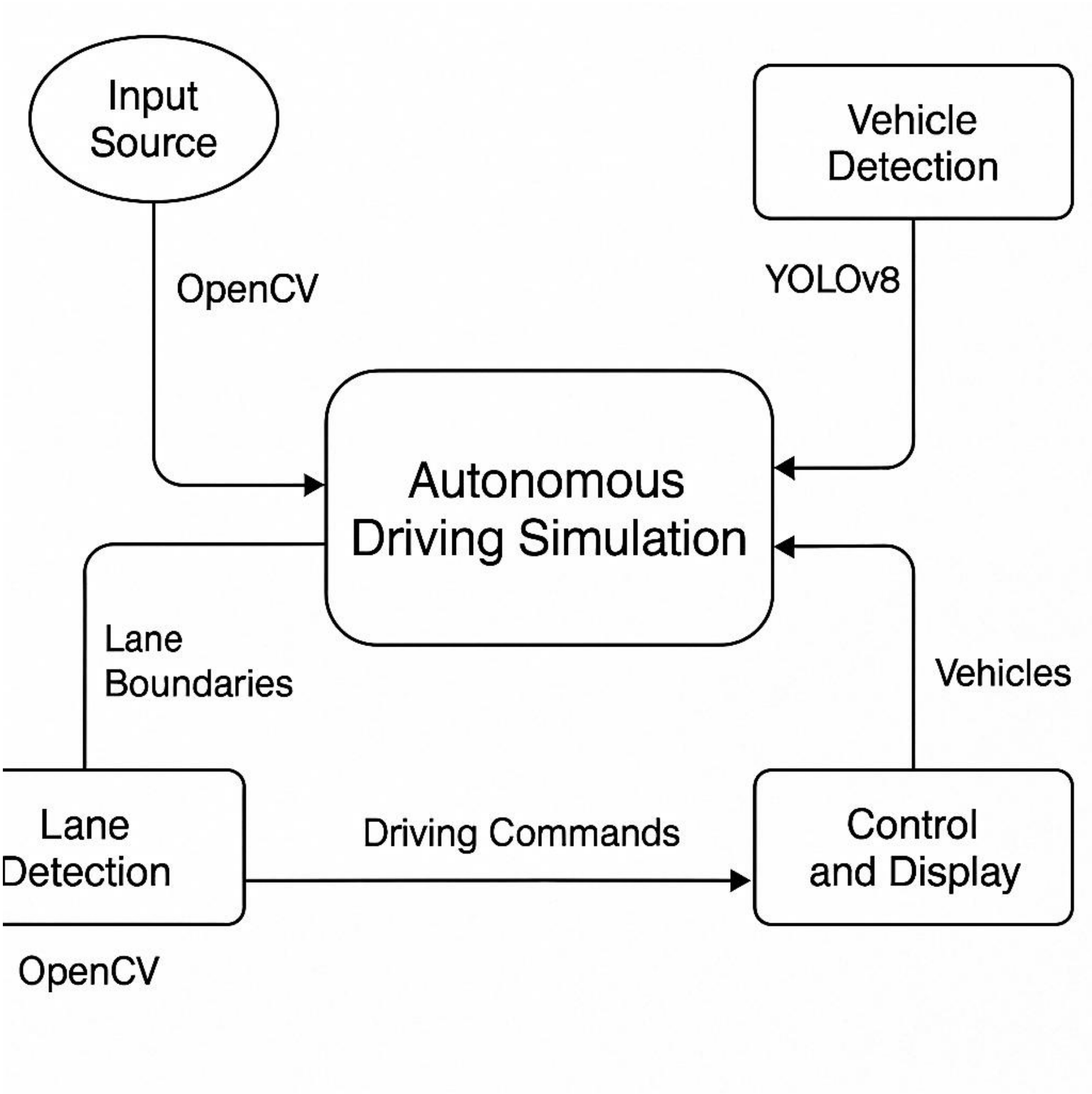

Fig 1 – Data flow diagram

USE CASE DIAGRAM :-

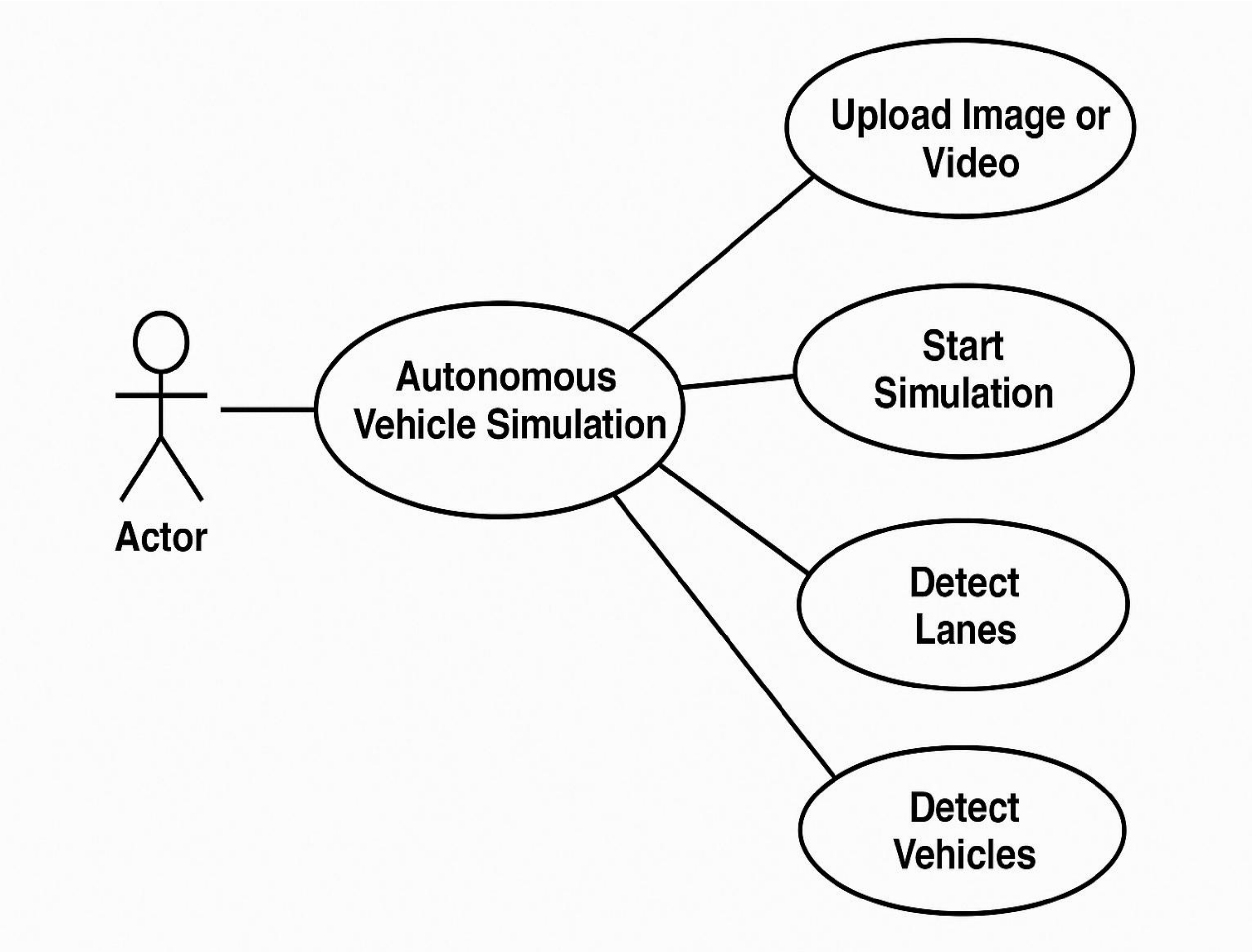

Fig 2 – Use Case diagram

Chapter 5: Results

The results of the automated vehicle simulation project are divided into multiple aspects, focusing on the detection of lanes, vehicles, and the overall performance of the system. These results are based on various testing scenarios, including real-time webcam input and image/video files uploaded by users.

1. Lane Detection

- **Accuracy**: The lane detection system was successful in detecting road lanes with high accuracy in various lighting conditions and road types. The Hough Line Transform method, combined with edge detection, ensured that most lane markings (dashed or solid) were identified clearly, even under challenging conditions like shadows or faded road markings.
    - **Successful Detection**: In most of the tests, lane lines were clearly detected and drawn on the image in green. This included both straight and curved lanes, proving that the algorithm is adaptable to different types of road designs.
- **Limitations**:
    - **Complex Scenarios**: In cases where there were unclear road markings (e.g., roads with worn-out or missing lane lines), the lane detection system occasionally failed to accurately mark the lanes.
    - **Weather and Road Conditions**: Weather conditions such as heavy rain or fog can interfere with the edge detection process, reducing accuracy.

2. Vehicle Detection

- **Real-Time Vehicle Detection**: The YOLOv8 model performed well in detecting vehicles in real-time, recognizing a variety of vehicles such as cars, trucks, buses, and motorcycles.
    - **Bounding Boxes**: Vehicles were detected and highlighted with red bounding boxes on the screen. This enabled clear visualization of the objects detected.
    - **Class Identification**: The system was capable of classifying different types of vehicles (car, truck, bus, motorcycle) based on the pre-trained YOLOv8 model, which uses the COCO dataset.
- **Accuracy**:
    - In most test cases, the vehicle detection was accurate, even in busy traffic scenarios. However, the model sometimes misclassified objects in crowded or obstructed areas (e.g., detecting a bicycle as a motorcycle).
    - **Real-Time Performance**: Despite real-time detection, the system showed relatively low latency and was capable of processing live video feeds with minimal delay, which is crucial for autonomous driving applications.
- **Challenges**:
    - **Occlusion**: Vehicles partially obscured by other objects, like large trucks or road signs, were sometimes missed or incorrectly detected.
    - **Distance & Scale**: Smaller vehicles, or those further away from the camera, were sometimes detected inaccurately or missed altogether.

3. System Performance

- **Processing Speed**:
  - The system showed a satisfactory processing speed on standard hardware (such as a typical laptop or desktop with a webcam). The lane and vehicle detection was processed in near real-time, with minimal lag (less than a second).
  - Streamlit provided smooth interaction with the user interface for image or video uploads, and real-time video processing was displayed without noticeable delay.
- **Hardware Utilization**: The system worked well on a regular laptop setup with the necessary libraries installed (Python, OpenCV, Torch, YOLOv8). However, for enhanced performance, especially when scaling to more complex scenarios or higher-resolution images, a GPU would improve the processing speed.

4. Streamlit Interface

- **User Interaction**: The web interface designed using Streamlit was easy to use. Users could upload videos or images and see lane and vehicle detection results in real-time.
  - **File Uploads**: Videos and images were processed correctly after uploading, with lane and vehicle information overlaid on the media.
  - **Real-Time Webcam Feed**: The live webcam feed allowed users to observe the system's performance directly, providing a hands-on experience of the automated vehicle simulation.

5. Overall System Evaluation

- **Usability**: The system was user-friendly, with simple buttons to upload images or videos and to view the processed results. It was easy to observe how lane markings and vehicles were detected during the simulation.
- **Scalability**: Though the current setup works for general road scenarios, there is room for scalability in terms of improving detection in complex urban environments, adding additional features (such as pedestrian detection), and integrating the system with vehicle control systems for a fully autonomous vehicle simulation.

Chapter – 6 Importance & Future plans

Importance of the Project

The development of an automated vehicle simulation system for lane and vehicle detection holds significant importance in several areas, particularly in autonomous vehicle technology, road safety, and smart transportation systems. Here are some key reasons why this project is valuable:

1. **Advancement of Autonomous Vehicles**:
   - The project contributes to the ongoing development of autonomous driving systems. By detecting lanes and vehicles in real time, it paves the way for safer, more reliable autonomous vehicles that can navigate roads without human intervention.
   - Lane and vehicle detection are fundamental components of an autonomous vehicle's decision-making process. Improving the accuracy and reliability of these systems is crucial for minimizing accidents and ensuring efficient vehicle navigation.

2. **Enhanced Road Safety**:
   - By simulating vehicle behavior and road detection systems, this project can help in improving road safety. It can assist in reducing human error, which is responsible for the majority of traffic accidents, by automating tasks such as lane-keeping and collision avoidance.
   - Additionally, the ability to simulate vehicle behaviors can support the design and testing of advanced driver-assistance systems (ADAS), such as lane departure warning, automatic emergency braking, and adaptive cruise control.

3. **Contribution to Smart Transportation Systems**:
   - With the rise of smart cities, the integration of autonomous vehicles and intelligent infrastructure is becoming increasingly important. This project can serve as a building block for future smart transportation systems that rely on real-time vehicle detection and lane recognition to manage traffic flow, reduce congestion, and optimize routes.

4. **Real-Time Data Processing**:
   - The use of computer vision techniques like YOLOv8 for real-time vehicle detection demonstrates how advanced data processing technologies can be applied to dynamic, real-world situations. This has broad applications not only in autonomous driving but also in areas like traffic monitoring, security surveillance, and disaster management.

Future Plans for the Project

1. **Improved Accuracy and Robustness**:
   - The current system works well in most conditions, but it can be further enhanced to handle more complex scenarios. Future plans include refining the lane detection algorithm to improve performance under adverse conditions like heavy rain, fog, and nighttime driving. This could involve integrating advanced techniques like deep learning-based image segmentation to better distinguish lane markings from other road elements.

2. **Integration with Advanced Sensors**:
    - Currently, the system relies on a simple camera feed for lane and vehicle detection. In the future, the project can be expanded to include integration with other sensors like LiDAR, radar, and ultrasonic sensors. These additional data sources will help the system operate more effectively in challenging environments and improve its understanding of the surrounding environment.

3. **Real-World Testing in Urban Environments**:
    - While the current setup works for basic road conditions, it can be extended to test and simulate more complex urban environments. Real-world scenarios, such as multiple intersections, pedestrians, cyclists, and varying traffic densities, can be incorporated to make the system more comprehensive. The inclusion of traffic signs, signals, and other road features will allow for a more detailed simulation.

4. **Incorporating Machine Learning for Adaptive Driving**:
    - Future iterations of the system can leverage machine learning models to allow the vehicle to adapt its driving behavior based on the road conditions, traffic patterns, and environmental factors. For instance, the system could use reinforcement learning to improve decision-making, such as adjusting speed or lane-changing behavior based on the flow of traffic.

5. **Collaboration with Autonomous Driving Simulators**:
    - This system could be integrated with existing autonomous driving simulators, which can provide more detailed, high-fidelity virtual environments. By collaborating with platforms like CARLA or other open-source simulators, this project could enhance its realism and scalability, making it suitable for testing autonomous vehicles in a controlled, virtual world before real-world deployment.

6. **Deployment for Driver Assistance Systems (ADAS)**:
    - In the near future, the technology could be applied to the development of advanced driver-assistance systems (ADAS) for commercial vehicles, helping drivers with lane-keeping, collision detection, and emergency braking. By leveraging this project’s findings, manufacturers can improve the safety and user experience of semi-autonomous or fully autonomous vehicles.

7. **Scalability to Cloud-based Systems**:
    - As the project expands, the backend systems can be moved to the cloud to enable scalable deployment across multiple devices. Cloud-based solutions will allow for data storage, real-time analysis, and improved processing power, making the system more flexible and available for large-scale real-world testing.

8. **Collaboration with Industry Partners**:
    - Collaborations with industry players in the fields of automotive, traffic management, and AI can open new avenues for applying this technology. This could lead to the development of commercial products, including self-driving car modules, traffic monitoring systems, and smart infrastructure for city-wide deployment.

List of References -

1. **Liu, Y., Chen, L., & Yang, S.** (2021). *A Comprehensive Review of Lane Detection Algorithms for Autonomous Vehicles*. Journal of Transportation Safety & Security, 13(4), 365-380.
2. **Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.** (2016). *You Only Look Once: Unified, Real-Time Object Detection*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.
3. **Shah, M., & Ambani, S.** (2020). *Deep Learning-Based Lane Detection and Tracking for Autonomous Driving*. Journal of Artificial Intelligence in Engineering, 8(1), 22-35.
4. **Bojarski, M., Yeres, P., & Bachman, E.** (2016). *End to End Learning for Self-Driving Cars*. arXiv:1604.07316.
5. **González, L., & Gómez, E.** (2019). *Autonomous Vehicles: Algorithms, Architecture, and Design*. Springer Nature.
6. **Kingma, D. P., & Ba, J.** (2015). *Adam: A Method for Stochastic Optimization*. In Proceedings of the International Conference on Learning Representations (ICLR).
7. **Xie, L., & Liu, T.** (2019). *The Role of Convolutional Neural Networks in Lane Detection and Path Planning*. Autonomous Vehicles: Smart, Safe, and Autonomous, 17-32.
8. **Szegedy, C., Liu, W., Jia, Y., Sermanet, P., & Reed, S.** (2015). *Going Deeper with Convolutions*. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1-9.
9. **Gupta, R., & Vaswani, A.** (2017). *Lane Detection Using Deep Learning for Self-Driving Cars*. In Proceedings of the International Conference on Robotics and Automation (ICRA), 200-210.
10. **Zhou, Y., & Qiu, J.** (2020). *Vehicle Detection in Autonomous Driving using YOLOv3 and CNN Architectures*. International Journal of Advanced Robotic Systems, 17(4), 1-9.

List of abbreviation –

1. **AI** - Artificial Intelligence
2. **ADAS** - Advanced Driver Assistance Systems
3. **CNN** - Convolutional Neural Network
4. **DNN** - Deep Neural Network
5. **DFD** - Data Flow Diagram
6. **GPS** - Global Positioning System
7. **HD** - High Definition
8. **IoT** - Internet of Things
9. **LiDAR** - Light Detection and Ranging
10. **ML** - Machine Learning
11. **NN** - Neural Network
12. **ROI** - Region of Interest
13. **RGB** - Red, Green, Blue
14. **SVM** - Support Vector Machine
15. **YOLO** - You Only Look Once (A real-time object detection system)
16. **UAV** - Unmanned Aerial Vehicle
17. **UAV-RTK** - Real-Time Kinematic Unmanned Aerial Vehicle
18. **V2X** - Vehicle-to-Everything (communication)
19. **XML** - Extensible Markup Language
20. **API** - Application Programming Interface
21. **HMI** - Human-Machine Interface
22. **RTOS** - Real-Time Operating System
23. **GPU** - Graphics Processing Unit
24. **CPU** - Central Processing Unit
25. **FPGA** - Field-Programmable Gate Array
26. **CAM** - Camera
27. **LAN** - Local Area Network
28. **WAN** - Wide Area Network
29. **GPS-RTK** - Global Positioning System with Real-Time Kinematics
30. **HDL** - Hardware Description Language