COLLEGE CODE: 3108

COLLEGE NAME:JEPPIAAR ENGINEERING COLLEGE

DEPARTMENT: INFORMATION TECHNOLOGY

STUDENT NM-ID:

A5A133D9A281447A53445EE126597A4F

ROLL NO: 310823205004

DATE: 11-05-2025

Completed the project named as

AI-EBPL -AUTONOMUS VEHICLES AND ROBOTICS

SUBMITTED BY:

NAME: AMI SAPTHIKA C

MOBILE NO: 8754909741

PHASE 3: IMPLEMENTATION OF PROJECT

USERCASE: AUTONOMOUS VEHICLES AND ROBOTICS

TITLE: TRAFFIC SIGN RECOGNITION USING AI

OBJECTIVE

The objective of implementing Traffic Sign Recognition using AI in Autonomous Vehicles and Robotics is to develop an accurate traffic sign recognition system that can detect, recognize, and classify traffic signs in real-time. This aims to improve road safety by enabling autonomous vehicles to obey traffic rules and regulations, increase efficiency by optimizing traffic flow and reducing congestion, and enhance autonomous vehicle performance by integrating accurate traffic sign recognition capabilities. Key performance indicators include achieving an accuracy rate of 95% or higher, processing data in real-time with less than 100ms latency, and ensuring robustness in various environmental conditions.

1.AI MODEL DEVELOPMENT

Overview

The AI model development involves designing and training a deep learning model to recognize and classify traffic signs accurately. The model will be integrated into autonomous vehicles to enhance road safety and efficiency.

Implementation

1. Data Collection: Gather a large dataset of images and videos of traffic signs from various sources.

2. Data Preprocessing: Clean, label, and preprocess the data for training.

3. Model Selection: Choose a suitable deep learning architecture (e.g., CNN, YOLO).

4. Model Training: Train the model using the preprocessed data.

5. Model Evaluation: Evaluate the model's performance using metrics such as accuracy, precision, and recall.

6. Model Optimization: Fine-tune the model for better performance.

Outcome

The outcome of the AI model development for Traffic Sign Recognition using AI in Autonomous vehicles and Robotics is the creation of an accurate and reliable system that can recognize and classify traffic signs in real-time. This leads to improved road safety as autonomous vehicles equipped with the AI model can obey traffic rules and regulations, reducing accidents. Additionally, the AI model enables autonomous vehicles to navigate through traffic efficiently, reducing congestion and increasing efficiency. The system can process traffic sign recognition data in real-time, enabling timely decision-making, and is robust and reliable in various environmental conditions, including weather, lighting, and occlusion.

2.CHATBOT DEVELOPMENT

Overview

The chatbot development involves designing and integrating a conversational interface that provides information and assistance to users regarding traffic sign recognition and autonomous vehicle operations.

Implementation

1. Chatbot Platform Selection: Choose a suitable chatbot development platform (e.g., Dialogflow, Microsoft Bot Framework).

2. Intent Identification: Identify user intents and develop conversational flows.

3. Dialogue Management: Design and implement dialogue management to handle user queries.

4. Integration with AI Model: Integrate the chatbot with the traffic sign recognition AI model.

5. Testing and Deployment: Test and deploy the chatbot on various platforms (e.g., web, mobile, voice assistants).

Outcome

The outcome of the chatbot development for Traffic Sign Recognition using AI in Autonomous vehicles and Robotics is the creation of a user-friendly conversational interface that provides convenient access to information about traffic signs and autonomous vehicle operations. The chatbot enhances the overall user experience, promotes safe driving practices by providing critical information and alerts, and increases efficiency by assisting users in navigating through traffic, reducing congestion.

3.IOT DEVICE INTEGRATION

Overview

The IoT device integration involves connecting and integrating various IoT devices, such as sensors, cameras, and GPS, to enable real-time data collection and exchange for traffic sign recognition and autonomous vehicle operations.

Implementation

1. Device Selection: Select suitable IoT devices (e.g., cameras, sensors, GPS) for data collection.

2. Device Integration: Integrate IoT devices with the traffic sign recognition AI model and autonomous vehicle systems.

3. Data Processing: Process and analyze real-time data from IoT devices for traffic sign recognition.

4. Communication Protocol: Establish communication protocols for data exchange between IoT devices and autonomous vehicle systems.

5. Security Measures: Implement security measures to ensure data integrity and prevent unauthorized access.

Outcome

The outcome of the IoT device integration for Traffic Sign Recognition using AI in Autonomous vehicles and Robotics is the enablement of real-time data collection and analysis, enhancing autonomous vehicle operations and improving road safety. The integration allows for prompt responses to safety-critical situations, optimizes traffic flow, reduces congestion, and increases efficiency.

4.DATA SECURITY IMPLEMENTATION

Overview

The data security implementation involves protecting sensitive data related to traffic sign recognition, autonomous vehicle operations, and user information from unauthorized access, theft, or manipulation.

Implementation

1. Data Encryption: Encrypt data both in transit and at rest using secure protocols (e.g., TLS, AES).

2. Access Control: Implement role-based access control and authentication mechanisms (e.g., username/password, biometrics).

3. Data Anonymization: Anonymize sensitive data to prevent identification of individuals or vehicles.

4. Secure Communication Protocols: Establish secure communication protocols for data exchange between autonomous vehicles, infrastructure, and cloud services.

5. Regular Security Audits: Conduct regular security audits and penetration testing to identify vulnerabilities.

Outcome

The outcome of the data security implementation for Traffic Sign Recognition using AI in Autonomous vehicles and Robotics is the assurance of confidentiality, integrity, and availability of sensitive data. The implementation maintains data integrity, prevents unauthorized access, and ensures compliance with relevant data protection regulations, ultimately fostering trust among users and stakeholders that their sensitive data is protected.

5.TESTING AND FEEDBACK COLLECTION

Overview

The testing and feedback collection involves evaluating the traffic sign recognition system's performance, identifying areas for improvement, and gathering feedback from stakeholders to ensure the system meets the required standards.

Implementation

1. Unit Testing: Conduct unit testing to verify individual components' functionality.

2. Integration Testing: Perform integration testing to ensure seamless interaction between components.

3. System Testing: Conduct system testing to evaluate the overall system's performance.

4. User Acceptance Testing (UAT): Perform UAT to gather feedback from stakeholders.

5. Feedback Collection: Collect feedback through surveys, interviews, or focus groups.

6. Iterative Testing: Conduct iterative testing to refine the system based on feedback

Outcome

The outcome of the testing and feedback collection for Traffic Sign Recognition using AI in Autonomous vehicles and Robotics is the improvement of the system's performance, user experience, and accuracy. Through iterative testing and feedback incorporation, the system's defects are identified and resolved, resulting in increased stakeholder satisfaction and compliance with relevant standards and regulations

CHALLENGES AND SOLUTIONS

1. Model Accuracy

Challenges: Limited training data, complexity of traffic signs, adverse weather conditions.

Solutions: Data augmentation, transfer learning, ensemble methods.

2.User Experience

Challenges: Complexity of AI technology, trust and reliability, information overload.

Solutions: Intuitive interface, transparency and explainability, personalized feedback.

3. IoT Device Availability

Challenges: Device connectivity, maintenance, scalability.

Solutions: Cellular/LPWAN connectivity, OTA updates, edge computing.

OUTCOMES OF PHASE 3

1. Improved Road Safety: Accurate traffic sign recognition reduces accidents and enhances road safety.

2. Enhanced Autonomous Vehicle Performance: AI-powered traffic sign recognition enables autonomous vehicles to navigate roads more efficiently.

3. Increased Efficiency: Reduced congestion and improved traffic flow due to accurate traffic sign recognition.

4. Reduced Manual Intervention: Autonomous vehicles can operate with minimal human intervention, reducing driver fatigue.

5. Data-Driven Insights: Collection of data on traffic patterns, sign recognition, and autonomous vehicle performance for future improvements.

NEXT STEPS FOR PHASE 4

Short-Term (0-6 months)

1. System Deployment: Deploy the traffic sign recognition system in a controlled environment.

2. Testing and Validation: Conduct thorough testing and validation of the system.

3. Pilot Program: Launch a pilot program with a small fleet of autonomous vehicles.

Mid-Term (6-18 months)

1. System Refining: Refine the system based on feedback from the pilot program.

2. Scalability Testing: Test the system's scalability with a larger fleet of autonomous vehicles.

3. Integration with Other Systems: Integrate the traffic sign recognition system with other autonomous vehicle systems.

Long-Term (18+ months)

1. Commercial Deployment: Deploy the system commercially with autonomous vehicle manufacturers.

2. Continuous Improvement: Continuously collect data and improve the system's accuracy and efficiency.

3. Expansion to New Markets: Explore new markets and applications for the traffic sign recognition system.

PROGRAM

```python

import numpy as np
import cv2
import time
import pathlib


"""
a blueprint for a bounded box with its corresponding name,confidence score and
"""
print(pathlib.Path.cwd())

class BoundedBox:

    Tabnine | Edit | Test | Explain | Document
    def __init__(self, xmin, ymin, xmax, ymax, ids, confidence):
        with open(str(pathlib.Path.cwd().parents[0])+"/datas/coco.names", 'rt') as f:
            self.classes = f.read().rstrip('\n').split('\n')  # stores a list of classes
        self.xmin = xmin
        self.ymin = ymin
        self.xmax = xmax
        self.ymax = ymax
        self.name = self.classes[ids]
        self.confidence = confidence


"""
a blueprint that has lanes as lists and give queue like functionality
to reorder lanes based on their turn for green and red light state
"""
```

```python
class Lanes:
    Tabnine | Edit | Test | Explain | Document
    def __init__(self,lanes):
        self.lanes=lanes

    Tabnine | Edit | Test | Explain | Document
    def getLanes(self):

        return self.lanes

    Tabnine | Edit | Test | Explain | Document
    def lanesTurn(self):

        return self.lanes.pop(0)

    Tabnine | Edit | Test | Explain | Document
    def enque(self,lane):

        return self.lanes.append(lane)
    Tabnine | Edit | Test | Explain | Document
    def lastLane(self):
        return self.lanes[len(self.lanes)-1]
"""
a blueprint that has lanes as lists and give queue like functionality
to reorder lanes based on their turn for green and red light state
"""
class Lane:
    Tabnine | Edit | Test | Explain | Document
    def __init__(self,count,frame,lane_number):
        self.count = count
        self.frame = frame
        self.lane_number = lane_number

"""
```

```
project
|    README.md
|    requirement.txt
|
|__  common
|    |  utils.py
|__  datas
     |  video.mp4
     |  video1.mp4
     |  video2.mp4
     |  video3.mp4
     |  video4.mp4
     |  coco.name
|__  implementation_with_yolov5s_onnx_model
     |  main.py
|__  implementation_with_yolov5s_tensorrt_model
     |  processor.py
     |  main.py
|__  models
     |  yolov5s.onnx
     |  yolov5s.trt
```

SAMPLE OUTPUT: