REAL-TIME COLLISION PREVENTION SYSTEM FOR HAIRPIN BENDS USING ESP32-CAM AND OPEN CV

A DESIGN PROJECT REPORT

Submitted by

SASIDARAN K

VISHWA P

VELMURUGAN S

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2024

REAL-TIME COLLISION PREVENTION SYSTEM FOR HAIRPIN BENDS USING ESP32-CAM AND OPEN CV

A DESIGN PROJECT REPORT

Submitted by

SASIDARAN K (811721104092)

VISHWA P (811721104124)

VELMURUGAN S (811721104307)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **“REAL TIME COLLISION PREVENTION SYSTEM FOR HAIRPIN BENDS USING ESP32 CAM AND OPENCV”** is the bonafide work of **SASIDARAN K (811721104092), VISHWA P (811721104124), VELMURUGAN S (811721104307)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or anyother candidate.

SIGNATURE	**SIGNATURE**
Dr. A. Delphin Carolina Rani M.E., Ph.D.,	Mr. P. Matheswaran M.E.,(Ph.D.,)
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Department of CSE	Department of CSE
K. Ramakrishnan College of Technology	K. Ramakrishnan College of Technology
(Autonomous)	(Autonomous)
Samayapuram – 621 112	Samayapuram – 621 112

Submitted for the viva-voce examination held on …………….

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION

We jointly declare that the project report on **"REAL TIME COLLISION PREVENTION SYSTEM FOR HAIRPIN BENDS USING ESP32 CAM AND OPEN CV"** is the result of original work done by us and bestof our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the award of Degree of **BACHELOR OF ENGINEERING**.

Signature

____

SASIDARAN K

____

VISHWA P

____

VELMURUGAN S

Place: Samayapuram

Date:

ACKNOWLEDGEMENT

It is with great pride that we express our gratitude and indebtedness to our institution, “**K. Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

We extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided the facilities during the course of our study in college.

We would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

We would like to thank **Dr. N. VASUDEVAN, M.E., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

We thank **Dr. A. DELPHIN CAROLINA RANI M.E., Ph.D.,** Head of the Department of **COMPUTER SCIENCE AND ENGINEERING,** for providing her encouragement in pursuing this project.

We wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mr. P. MATHESWARAN M.E., (Ph.D).,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for her incalculable suggestions, creativity, assistance andpatience, which motivated us to carry out this project.

We render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

We wish to express our special thanks to the officials and Lab Technicians ofour departments who rendered their help during the period of the work progress.

ABSTRACT

A real-time collision prevention system specifically designed for hairpin bends, utilizing the ESP32-CAM microcontroller for wireless video transmission to a laptop equipped with OpenCV and a vehicle identification model. The system aims to enhance road safety by detecting vehicles approaching from the opposite direction in hairpin bends, where visibility is often limited. By leveraging continuous video monitoring and analysis, the system identifies potential collision risks and provides timely alerts to drivers, notifying them immediately when a vehicle is detected approaching the hairpin bend. This real-time notification allows drivers to take necessary precautions, helping to prevent accidents and ensure safer navigation through challenging road conditions. The system addresses the unique challenges of hairpin bends, significantly improving driver awareness and safety. Its robust and scalable design adapts to various road conditions and future technological advancements, offering a tailored solution where traditional safety measures may fall short. Overall, the system delivers accurate, timely information about potential collision risks, reducing the likelihood of accidents and promoting safer driving practices in complex road environments.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

MQTT	-	Message Query Telementary Transport.
CNN	-	Convolutional Neural Network
YOLO	-	You Only look Once
ESP32-CAM	-	Espressif Systems Processor 32 Camera
SVM	-	Support Vector Machine
ADAS	-	Advanced Driver Assistance Systems

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

An innovative real-time collision prevention system tailored specifically for hairpin bends, where the combination of sharp turns and limited visibility presents significant challenges to drivers. Leveraging cutting-edge technology, the system utilizes the ESP32-CAM microcontroller for wireless video transmission and OpenCV on a laptop for real-time video analysis. The system's primary objective is to enhance road safety by detecting vehicles approaching from the opposite direction in hairpin bends and providing timely alerts to drivers to prevent potential collisions. At its core, the ESP32-CAM serves as the hardware backbone, capturing real-time video footage of the hairpin bend, while OpenCV processes this video feed to detect and identify vehicles using a pre-trained machine learning model. The system's architecture ensures seamless interaction between hardware and software components, allowing for continuous monitoring and analysis of the road conditions. Implementation involves meticulous setup and configuration to optimize the system's performance, including fine-tuning camera settings, establishing a stable Wi-Fi connection, and integrating the vehicle identification model with OpenCV. Extensive scenario testing validates the system's effectiveness across various hairpin bend conditions, assessing detection accuracy, real-time processing capabilities, and alert responsiveness. Despite promising performance, the system faces challenges such as environmental factors and network stability, which necessitate ongoing optimization and enhancements. Future improvements will focus on enhancing detection accuracy, optimizing performance, and expanding alert mechanisms to ensure safer navigation through hairpin bends.

1.2 PROBLEM STATEMENT

Hairpin bends on roads present a significant safety challenge due to their sharp turns and limited visibility, often resulting in accidents with severe consequences. The existing road infrastructure lacks adequate measures to address the specific risks posed by these bends, leading to a pressing need for advanced collision prevention systems tailored specifically for hairpin bends. The primary problem lies in the lack of real-time detection and alert mechanisms to warn drivers of approaching vehicles from the opposite direction in hairpin bends, where visibility is severely restricted. Traditional safety measures such as signage and road markings are often insufficient in mitigating the risks associated with hairpin bends, highlighting the necessity for innovative solutions that leverage technology to enhance safety in these areas. Therefore, the problem addressed in this paper is the development of a real-time collision prevention system designed specifically for hairpin bends, aiming to detect potential collision risks and provide timely alerts to drivers to prevent accidents and ensure safer navigation through challenging road conditions.

1.3 OBJECTIVES

The primary objective of the proposed real-time collision prevention system for hairpin bends is to enhance road safety by detecting and alerting drivers to potential collisions. By improving visibility in hairpin bends, the system aims to make drivers aware of oncoming traffic where visibility is often limited. Utilizing the ESP32-CAM microcontroller and deep learning models, the system ensures accurate and real-time vehicle detection under various conditions. Predictive analysis algorithms assess collision risks based on detected vehicle trajectories, enabling timely and meaningful alerts. The system features a user-friendly interface that provides clear, timely, and non-intrusive warnings, helping drivers take appropriate actions to avoid collisions. Additionally, the system is designed to be scalable, adaptable to various road conditions, and cost-effective to ensure broad adoption. Integration with existing infrastructure and potential future Vehicle-to-Everything (V2X) communication is a key objective. Continuous monitoring, maintenance, and updates are established to maintain effectiveness and reliability. Comprehensive testing and validation in controlled and real-world environments ensure the system meets safety standards and user expectations, ultimately reducing the risk of collisions in hairpin bends.

1.4 IMPLICATION

The implications of the proposed real-time collision prevention system for hairpin bends are significant and far-reaching. By enhancing visibility and situational awareness in challenging road conditions, the system has the potential to drastically reduce the incidence of collisions, thereby improving overall road safety. This reduction in accidents can lead to fewer injuries and fatalities, particularly in areas with numerous hairpin bends and limited visibility. Additionally, the deployment of such technology can result in lower emergency response costs and reduced insurance claims, benefiting both individuals and society at large. The system's cost-effectiveness and ease of integration with existing infrastructure make transforming road safety standards globally. Furthermore, the continuous monitoring and predictive capabilities of the system can provide valuable data insights, effective traffic management strategies.

1.5 SCOPE OF THE PROJECT

- The project involves the development and implementation of a robust hardware and software infrastructure, including the selection and integration of components such as the ESP32-CAM microcontroller, Wi-Fi modules, and deep learning models for vehicle detection.
- The scope extends to data collection and preprocessing, involving the gathering of video footage from hairpin bends, data labeling for training models, and preprocessing steps to prepare the data for a analysis.
- The project involves the development of advanced algorithms for real-time vehicle detection, motion tracking.

CHAPTER 2

LITERATURE SURVEY

2.1 Title: "Real-Time Vehicle Detection System”

Author: Yang, Cheng, and Wang

Year: 2019

This comprehensive survey delves into real-time vehicle detection and tracking techniques, pivotal components for driver assistance systems. The paper meticulously examines various methodologies encompassing traditional approaches and contemporary deep learning techniques. By elucidating the state-of-the-art methodologies, it offers valuable insights into the advanced methods employed in similar systems, thereby enriching the understanding of real-time object detection and tracking.

2.2 Title: "Real-time Traffic Sign Detection, Recognition, and Tracking”.

Author: Gade and Moeslund

Year: 2014

Focused on real-time traffic sign detection, recognition, and tracking, this survey addresses aspects pertinent to detecting and identifying vehicles, akin to the objectives of the proposed collision prevention system. It delves into the intricacies of real-time object detection and tracking methodologies, shedding light on the challenges and advancements in this domain, thereby contributing valuable insights to the proposed system's development.

2.3 Title: "Wireless Video Transmission for Surveillance Applications".

Author: Al-Anbuky et al.

Year: 2018

This survey explores wireless video transmission techniques, crucial for transmitting video footage captured by the ESP32-CAM microcontroller in the proposed collision prevention system. By evaluating various wireless transmission methods, it provides a comprehensive understanding of their advantages and limitations. This knowledge serves as a foundational resource for informing the system's design and implementation strategies.

2.4 Title: "Machine Learning for Autonomous Vehicles".

Author: Chen et al.

Year: 2015

This comprehensive survey explores the realm of machine learning techniques applied in autonomous vehicle systems, including vehicle detection and identification. By examining various machine learning algorithms and approaches, it provides a nuanced understanding of the design and implementation of vehicle identification models, pivotal for the proposed collision prevention system. This survey acts as a valuable resource, enriching the understanding of machine learning methodologies pertinent to the system's objectives.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing systems for addressing safety in hairpin bends primarily rely on traditional measures like signage and speed limits. Some may include basic surveillance cameras, but they lack real-time analysis capabilities. Advanced driver assistance systems (ADAS) and autonomous vehicle technologies offer some indirect safety benefits, but they aren't tailored specifically for hairpin bends. Collaborative safety systems using wireless communication exist but may struggle with accurate hazard detection in these challenging road structures. Overall, there's a need for specialized solutions like the proposed real-time collision prevention system to enhance safety in hairpin bends.

3.2 DISADVANTAGES

Reliability and Accuracy: The effectiveness of the system relies heavily on the accuracy and reliability of the video analysis and object detection algorithms. Variations in lighting conditions, adverse weather, and environmental factors could potentially impact the system's performance, leading to false positives or false negatives in collision risk detection.

Dependency on Network Connectivity: The wireless transmission of video data from the ESP32-CAM to the processing unit relies on stable Wi-Fi connectivity. Any disruptions or fluctuations in the network connection could affect the real-time performance of the system, potentially leading to delays or loss of critical data.

3.3 PROPOSED SYSTEM

The proposed real-time collision prevention system for hairpin bends represents a comprehensive approach to enhancing road safety in challenging driving conditions. At its core, the system integrates the ESP32-CAM microcontroller, equipped with a camera, to capture real-time video footage of hairpin bends. This footage is wirelessly transmitted to a processing unit, where advanced algorithms powered by the OpenCV computer vision library analyze the video feed to detect vehicles approaching from the opposite direction. Leveraging deep learning techniques, the system accurately identifies vehicles even in low visibility conditions, enabling it to assess potential collision risks in real-time. Upon detecting a collision risk, the system triggers timely alerts to the driver through visual and auditory signals, empowering them to take evasive action and prevent accidents.

By combining cutting-edge hardware and intelligent software algorithms, the proposed system offers proactive collision prevention tailored specifically for hairpin bends. Its ability to analyze vehicle trajectories and predict potential collision scenarios in real-time provides drivers with crucial information to navigate safely through challenging road conditions. With the potential to significantly reduce the risk of accidents and enhance overall road safety, the proposed system represents a promising advancement in collision prevention technology for hairpin bends, offering drivers greater confidence and security when traversing these complex road structures.

3.4 ADVANTAGES

- **Enhanced Safety**: By continuously monitoring hairpin bends in real-time, the system can detect potential collision risks before they escalate into accidents. This proactive approach to safety provides drivers with timely warnings, enabling them to take evasive action and avoid collisions, ultimately reducing the risk of accidents and injuries.
- **Tailored for Hairpin Bends**: Unlike generic safety systems, the proposed solution is specifically designed to address the unique challenges posed by

hairpin bends, where visibility is often limited and the risk of accidents is higher. By focusing on this specific road configuration, the system can provide targeted alerts and assistance that are tailored to the demands of navigating hairpin bends safely.

3.5 SYSTEM ARCHITECTURE

- The system architecture comprises three main components: data capture, processing, and alert delivery. The data capture component involves the ESP32-CAM microcontroller equipped with a camera to capture real-time video footage of hairpin bends. This footage is wirelessly transmitted to the processing unit, typically a laptop or PC, where the OpenCV library analyzes it for vehicle detection. Deep learning models enhance accuracy in detecting vehicles under various conditions. Predictive analysis algorithms anticipate collision risks based on vehicle trajectories. Upon detection of a potential collision, the system triggers visual and auditory alerts, empowering drivers to take evasive action. This architecture ensures seamless integration of hardware and software components to enhance road safety in hairpin bends. The project involves the development of advanced algorithms for real-time vehicle detection, motion tracking.

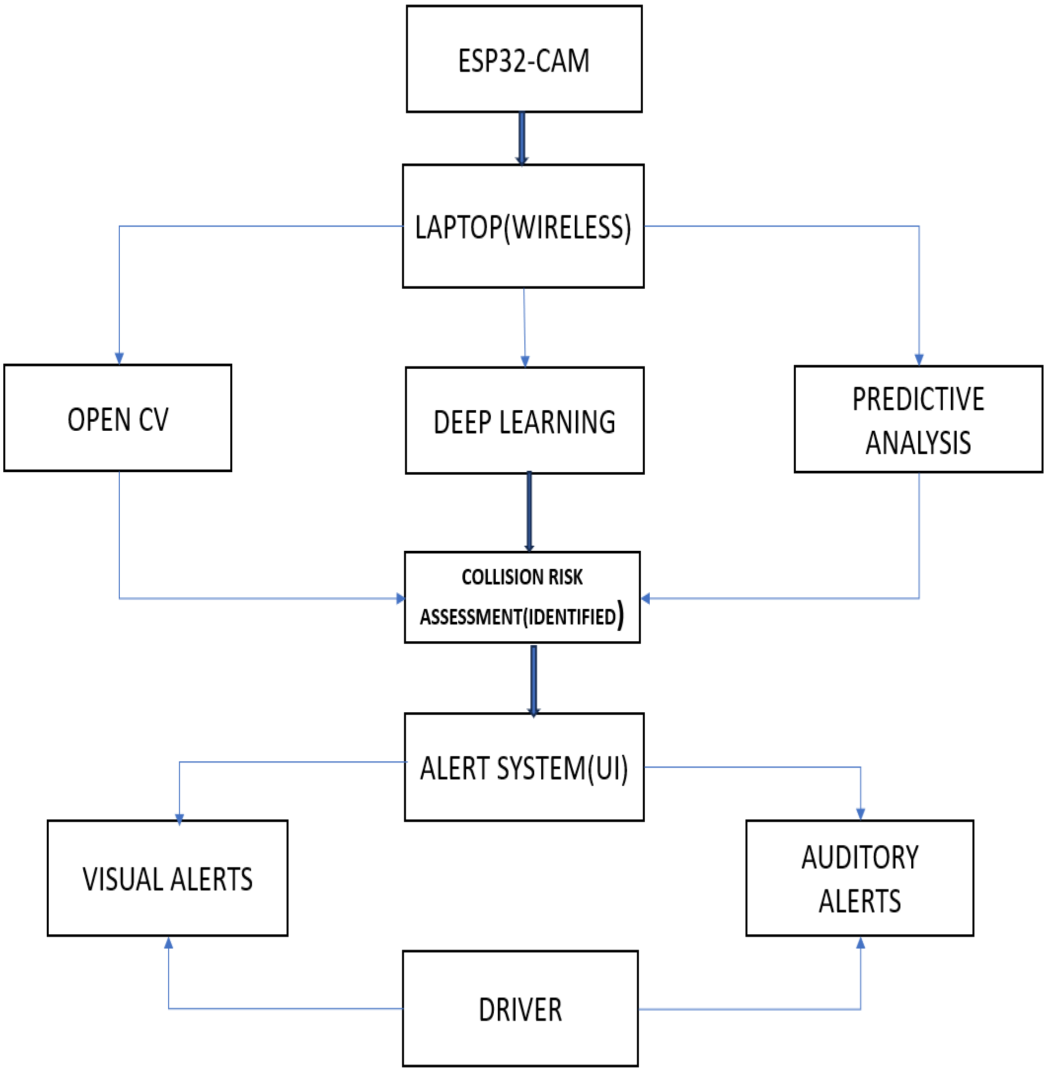

Fig 3.5 System Archietecture.

The project involves the development of advanced algorithms for real-time vehicle detection, motion tracking

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

- ESP32-CAM Microcontroller
- Laptop/PC
- Power supply
- Network Equipment:
  - Wi-Fi Router
  - Internet Connection

4.2 SOFTWARE REQUIREMENTS

Operating System - Windows 10 or 11 ,Macos, Linux.

- Development environment
  - Python 3x
  - Open cv
  - PyTorch
- Additional libraries and tools.
  - Numpy
  - Pandas
  - Django
  - Mqtt

4.3 SYSTEM INTERFACE

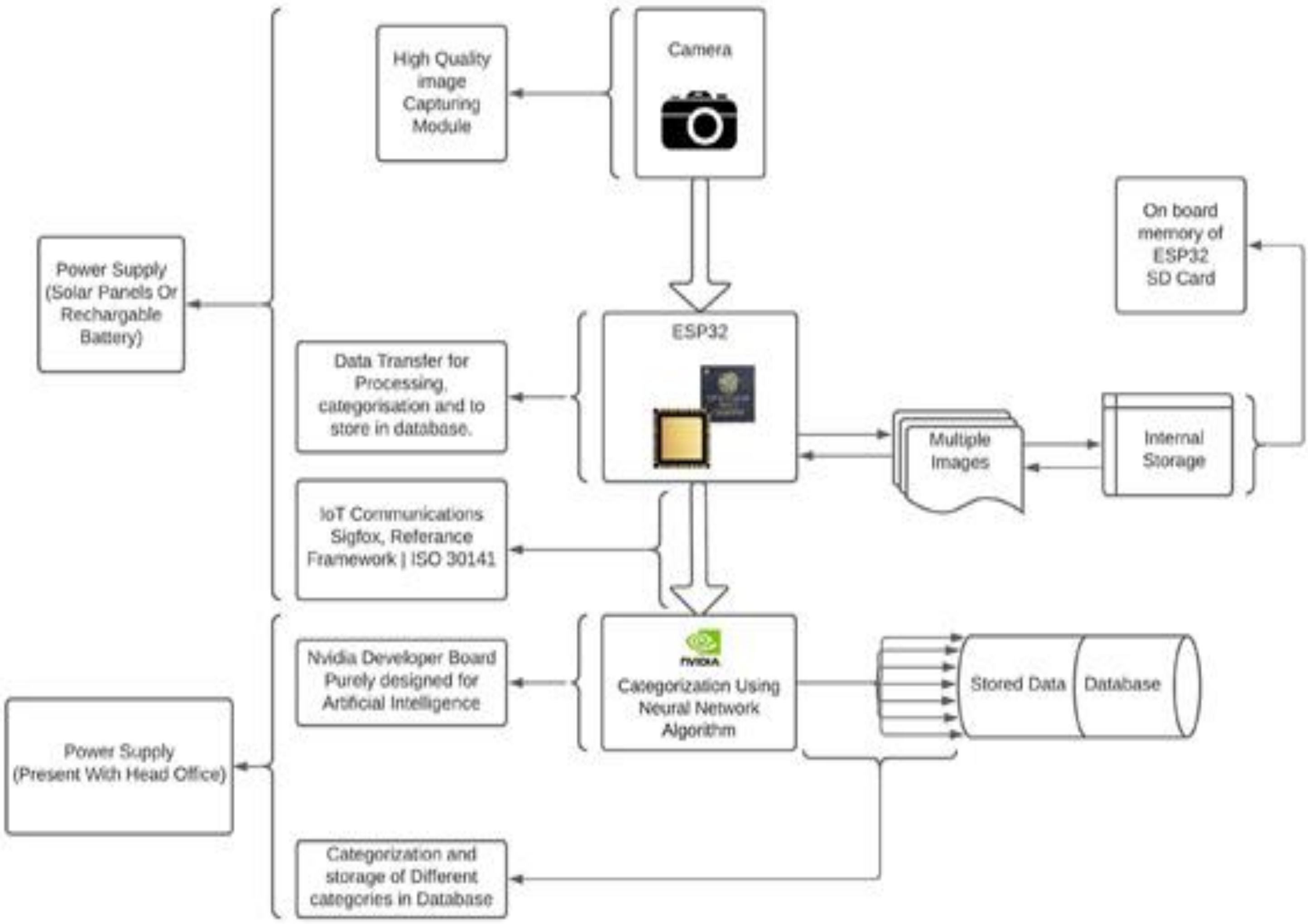

Fig. 4.3 System Interface

4.4 SOFTWARE DESCRIPTION

- **Real-time Video Processing**: Utilizes OpenCV for instantaneous capture and analysis of video feed from hairpin bends.
- **Deep Learning Models**: Integrates sophisticated deep learning models trained on labeled data for precise and reliable vehicle detection.

- **Motion Tracking**: Incorporates algorithms to track the movement of identified vehicles across consecutive frames.
- **Trajectory Prediction:** Implements predictive algorithms to anticipate the trajectories of vehicles, aiding in collision risk assessment.
- **User-Friendly Interface**: Features an intuitive interface designed for clear and prompt delivery of alerts to drivers.
- **Alert Optimization**: Utilizes intelligent algorithms to optimize the timing and context of alert notifications, minimizing driver distraction.
- **Communication Protocol:** Facilitates seamless communication between the ESP32-CAM microcontroller and the processing unit, enabling wireless video transmission and real-time analysis.
- **Monitoring and Maintenance:** Includes built-in functionalities for continuous monitoring and maintenance to ensure sustained reliability and effectiveness of the system..
- **Vehicle Identification Model**: A machine learning model trained to recognize vehicles within the video stream, enabling the system to differentiate between vehicles and other objects.

CHAPTER 5

MODULE DESCRIPTION

5.1 DATA COLLECTION MODULE AND INTEGRATION MODULE

This module is responsible for gathering and unifying diverse datasets necessary for training the vehicle detection and collision risk assessment models. Key activities include:

- **Source Identification:** Identify reliable sources of data such as video footage from road surveillance cameras, traffic reports, vehicle movement logs, and weather condition records.
- **Data Acquisition** Implement automated scripts and tools to collect data from identified sources, ensuring adherence to data privacy regulations.
- **Data Integration** Combine data from various sources into a unified dataset, addressing inconsistencies and ensuring data quality.

5.2 DATA PREPROCESSING MODULE

The preprocessing module prepares raw data for analysis and modeling by performing the following tasks:

- **Data Cleaning:** Remove or correct erroneous entries, handle missing values through imputation or deletion, and ensure data consistency.
- **Data Transformation:** Normalize or standardize numerical features, and encode categorical variables using techniques such as one-hot encoding or label encoding.
- **Feature Engineering:** Create new features from existing data to better capture the underlying patterns relevant to vehicle detection and collision risk assessment.

5.3 FEATURE SELECTION MODULE

This module focuses on selecting the most relevant features for the model to improve performance and reduce complexity:

- Correlation Analysis: Identify and remove highly correlated features to avoid redundancy.

- **Principal Component Analysis (PCA):** Reduce dimensionality by transforming features into a set of linearly uncorrelated components.
- **Recursive Feature Elimination (RFE):** Iteratively select features by fitting models and removing the least important features.

5.4 MACHINE LEARNING MODEL DEVELOPMENT MODULE

This module involves the implementation and training of various machine learning algorithms for vehicle detection and collision risk assessment:

- **YOLO (You Only Look Once):** A real-time object detection system that applies a single neural network to the full image, providing high-speed and accurate vehicle detection.
- **SSD (Single Shot MultiBox Detector):** Another real-time object detection model that discretizes the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location.
- **HOG (Histogram of Oriented Gradients):** A feature descriptor used in computer vision and image processing for the purpose of object detection.
- **SVM (Support Vector Machine):** A supervised learning model that analysis data for classification and regression analysis, useful for distinguishing vehicles from other objects.
- **Random Forest Classifier:** An ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.

5.5 MODEL EVALUATION MODULE

This module evaluates the performance of the trained models to ensure their reliability and effectiveness in vehicle detection and collision risk assessment:

- **Accuracy Score:** Measure the proportion of correct predictions made by the model.

- **Log Loss:** Evaluate the uncertainty of the predictions, penalizing incorrect predictions more heavily.

- **F1 Score:** Assess the balance between precision and recall, especially important in imbalanced datasets.

- **Confusion Matrix:** Visualize the performance by showing true positives, false positives, true negatives, and false negatives.

- **Cross-Validation:** Validate the model's performance using multiple splits of the dataset to ensure robustness.

5.6 USER INTERFACE MODULE

This module focuses on developing a user-friendly interface for drivers and system operators to interact with the collision prevention system:

- **Data Input:** Design intuitive forms and input fields for entering and adjusting system parameters, ensuring ease of use and accuracy.
- **Alert Output:** Display collision risk alerts clearly, including the location, time, and nature of the risk.
- **Visualization:** Incorporate charts, graphs, and other visual aids to help users understand the real-time detection results and underlying data patterns.

5.7 PREDICTION AND VISUALIZATION MODULE

This module generates and visualizes predictions, helping users interpret the results effectively:

- **Prediction Generation:** Use the trained models to predict vehicle detection and collision risks based on input data.

- **Visualization Tools:** Develop tools to display prediction results, such as real-time alerts, bar charts for traffic density, and heatmaps of risk areas.

5.8 DEPLOYMENT MODULE

This module handles the deployment of the collision prevention system in a real-world road environment:

- **Infrastructure Setup:** Establish the necessary hardware and software infrastructure, ensuring scalability and reliability.
- **Integration:** Integrate the system with existing traffic management systems and road safety databases.
- **Security:** Implement robust security measures to protect data and ensure compliance with road safety regulations.
- **F1 Score:** Assess the balance between precision and recall, especially important in imbalanced datasets.
- **Confusion Matrix:** Visualize the performance by showing true positives, false positives, true negatives, and false negatives.

CHAPTER 6

MACHINE LEARNING MODELS

6.1 CONVOLUTIONAL NEURAL NETWORK (CNN)

The Convolutional Neural Network (CNN) is utilized in this project for predicting collision risks at hairpin bends, leveraging its superior capability in image and video analysis. CNNs are particularly effective for this application due to their layered architecture, consisting of convolutional layers that automatically extract hierarchical features from input data, pooling layers that reduce dimensionality and computational load, and fully connected layers that synthesize the extracted features for final predictions. The CNN's ability to automatically learn and extract critical features from raw video footage significantly improves the accuracy and reliability of collision risk predictions, aiding in timely alerts and preventive measures for drivers. Overall, the CNN enhances road safety by providing real-time, data-driven insights into potential collision risks, ensuring safer navigation through challenging hairpin bends.

Fig. 6.1 Convolutional Neural Network

6.2 YOU ONLY LOOK ONCE (YOLO)

The You Only Look Once (YOLO) algorithm is employed in this project for predicting collision risks at hairpin bends, capitalizing on its real-time object detection capabilities. YOLO processes video data by applying a single neural network to the entire image, dividing it into regions, and predicting bounding boxes and probabilities for each region simultaneously. This approach allows YOLO to achieve high-speed detection without compromising accuracy, making it ideal for real-time applications where timely alerts are crucial. YOLO is trained on a comprehensive dataset that includes various features such as vehicle speed, trajectory, and environmental conditions extracted from road surveillance video footage. During the training phase, YOLO learns to detect vehicles accurately by optimizing the bounding box predictions and classification probabilities for each detected object. This enables the system to identify vehicles approaching from the opposite direction in hairpin bends, where visibility is often limited.

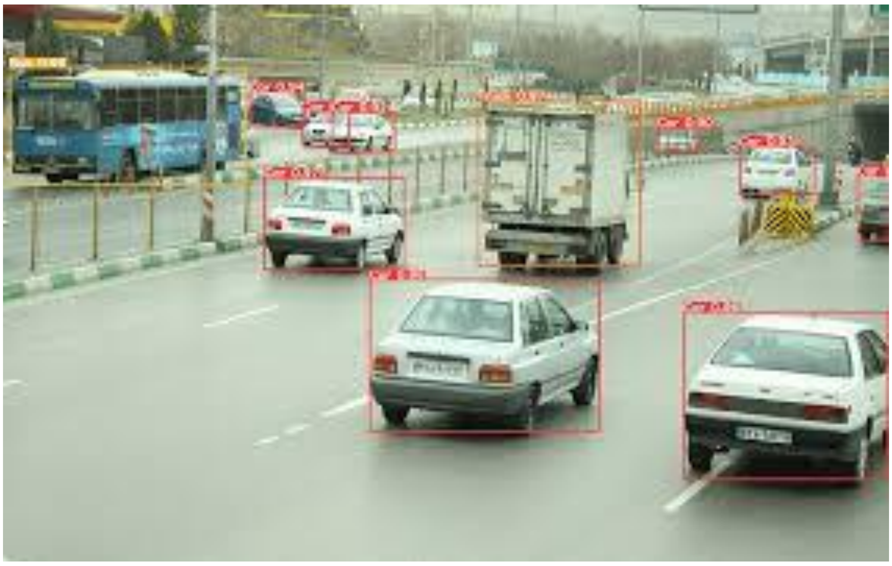

Fig. 6.2 You Only Look Once

6.3 SUPPORT VECTOR MACHINE (SVM)

The Support Vector Machine (SVM) classifier is used in this project to predict collision risks at hairpin bends. The model is trained on a dataset that includes features such as vehicle speed, trajectory, and environmental conditions, extracted from video footage and other sensors. After training, the SVM model makes predictions on new data to assess collision risks. The accuracy of these predictions is evaluated using metrics like accuracy score, log loss, and F1 score. This approach ensures reliable detection of potential collisions, aiding in timely alerts and enhancing road safety.

Fig. 6.3 Support Vector Machine (SVM)

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The proposed real-time collision prevention system for hairpin bends leverages advanced technologies to enhance road safety. Utilizing the ESP32-CAM microcontroller for real-time video capture and wireless transmission, the system monitors hairpin bends effectively. With OpenCV for computer vision and deep learning models for accurate vehicle detection, it reliably identifies approaching vehicles even in challenging conditions. Real-time processing and predictive analysis enable the system to assess collision risks by forecasting vehicle trajectories, providing timely visual and auditory alerts to drivers, allowing them to take immediate action to avoid accidents.

This system addresses the unique challenges of hairpin bends, significantly improving driver awareness and safety. Its robust and scalable design adapts to various road conditions and future technological advancements, offering a tailored solution where traditional safety measures may fall short. By delivering accurate, timely information about potential collision risks, the system not only reduces the likelihood of accidents but also promotes safer driving practices, making a substantial impact on road safety in complex road environments.

7.2 FUTURE ENHANCEMENT

While the proposed real-time collision prevention system for hairpin bends offers significant advancements in road safety, several future enhancements can be implemented to further improve its effectiveness and scalability:

Enhanced Deep Learning Models:

- Improved Training Data: Collect more extensive and diverse datasets, lighting scenarios, and road types, to improve the accuracy and robustness of the vehicle detection models.
- Model Optimization: Explore lightweight and efficient deep learning models

optimized for edge devices, reducing latency and computational requirements.

Advanced Sensor Fusion:

- Multi-Sensor Integration: Combine video data with other sensors like LiDAR, radar, and ultrasonic sensors to enhance environmental perception and provide a more comprehensive understanding of the surroundings.

Augmented Reality (AR) Integration:

- AR Displays: Use augmented reality displays within the vehicle to provide intuitive and non-intrusive visual alerts directly on the windshield, enhancing driver awareness without causing distractions.

Adaptive and Context-Aware Alerts:

- Customizable Alerts: Allow drivers to customize alert settings based on, such as adjusting the sensitivity and types of alerts.
- Context-Aware Notifications: Develop intelligent alert systems that consider the driving context, such as speed, traffic conditions, and driver behavior, to provide more relevant and timely warnings.

APPENDIX A
SOURCE CODE

```
import cv2
import urllib.request
import numpy as np
import pyttsx3
url = 'http://192.168.83.96/cam-hi.jpg'
winName = 'ESP32 CAMERA'
cv2.namedWindow(winName, cv2.WINDOW_AUTOSIZE)
classNames = []
classFile = 'coco.names'
with open(classFile, 'rt') as f:
    classNames = f.read().rstrip('\n').split('\n')
valid_classes = {
    'person': b'1',
    'car': b'2',
    'truck': b'3',
    'bus': b'4',
    'train': b'5',
    'motorcycle': b'6',
    'bicycle': b'7',
    'street sign': b'8',
    'stop sign': b'9',
    'fire hydrant': b'10'

}
configPath = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
        weightsPath = 'frozen_inference_graph.pb'
net = cv2.dnn_DetectionModel(weightsPath, configPath)
net.setInputSize(320, 320)
net.setInputScale(1.0 / 127.5)
```

```
net.setInputMean((127.5, 127.5, 127.5))
net.setInputSwapRB(True)
engine = pyttsx3.init()
while True:
    imgResponse = urllib.request.urlopen(url)
    imgNp = np.array(bytearray(imgResponse.read()), dtype=np.uint8)
    img = cv2.imdecode(imgNp, -1)
    img = cv2.rotate(img, cv2.ROTATE_90_CLOCKWISE)
    classIds, confs, bbox = net.detect(img, confThreshold=0.5)
    print(classIds, bbox)
    detected_objects = []
    if isinstance(classIds, np.ndarray):
        for classId, confidence, box in zip(classIds.flatten(), confs.flatten(), bbox):
            if classId - 1 < len(classNames):
                className = classNames[classId - 1]
                if className in valid_classes and confidence > 0.5:
                    detected_objects.append(className)
                    x, y, w, h = box
                    cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)
                    cv2.putText(img, className, (x, y - 10),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
                    engine.say(f"{className} detected!")
                    engine.runAndWait()
    obj_window_height = max(len(detected_objects) * 100, 100)
    obj_window = np.zeros((obj_window_height, 400, 3), dtype=np.uint8)
    for i, obj_name in enumerate(detected_objects):
        cv2.putText(obj_window, obj_name, (20, 30 * (i + 1)),
    cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 255, 255), 2)
    cv2.imshow("Detected Objects", obj_window)

    cv2.imshow(winName, img)
```

```
    key = cv2.waitKey(1)
    if key == 27:
        break

cv2.destroyAllWindows()
```

MACHINE LEARNING CODE

```
import cv2
import numpy as np

# Load YOLO
net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")
layer_names = net.getLayerNames()
output_layers = [layer_names[i - 1] for i in net.getUnconnectedOutLayers()]

# Load COCO names
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]

# Define the video stream URL from ESP32-CAM
url = "http://192.168.1.100:81/stream"  # Replace with your ESP32-CAM URL

# Initialize video capture
cap = cv2.VideoCapture(url)

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()
```

```
if not ret:
    print("Failed to grab frame")
    break

height, width, channels = frame.shape

# Detecting objects
blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
outs = net.forward(output_layers)

# Information to show on screen
class_ids = []
confidences = []
boxes = []

# For each detection from each output layer
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.5:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
```

```
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

    # Apply non-max suppression
    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)

    # Draw bounding boxes and labels on the frame
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            confidence = confidences[i]
            color = (0, 255, 0)
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
    center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[2] * width)
            h = int(detection[3] * height)
            # Rectangle coordinates
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)
    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

SUPPORT VECTOR MACHINE:

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, log_loss, f1_score, confusion_matrix

data = pd.read_csv('collision_data.csv')

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

svm_classifier = SVC(probability=True, random_state=42)
svm_classifier.fit(X_train, y_train)
y_pred = svm_classifier.predict(X_test)
y_prob = svm_classifier.predict_proba(X_test)
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
print(f"Accuracy: {accuracy}")
print(f"Log Loss: {logloss}")
print(f"F1 Score: {f1}")
print(f"Confusion Matrix:\n{conf_matrix}")
```

APPENDIX B
SCREENSHOTS

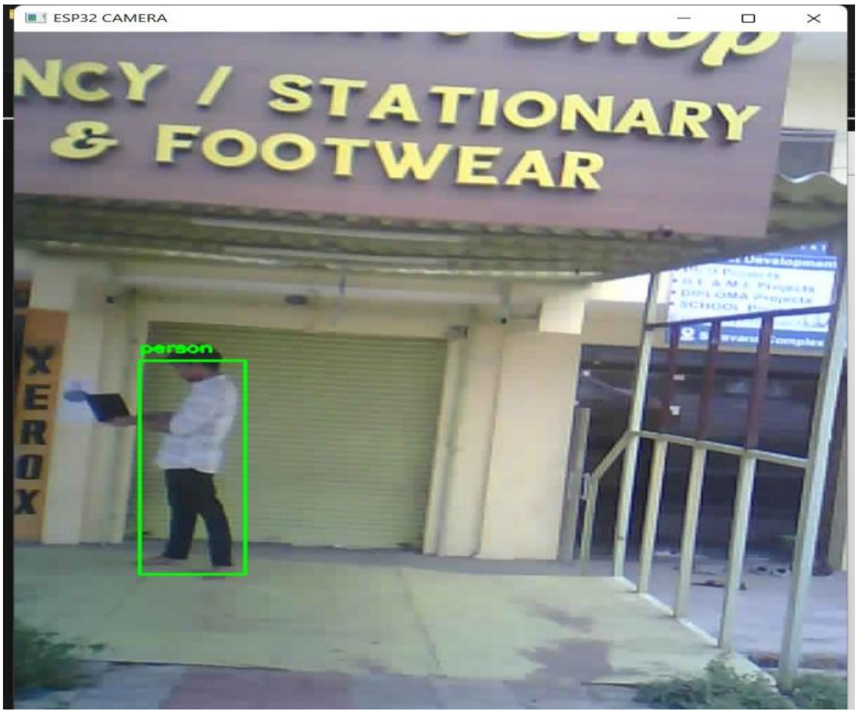

Fig. B1 Person Detection

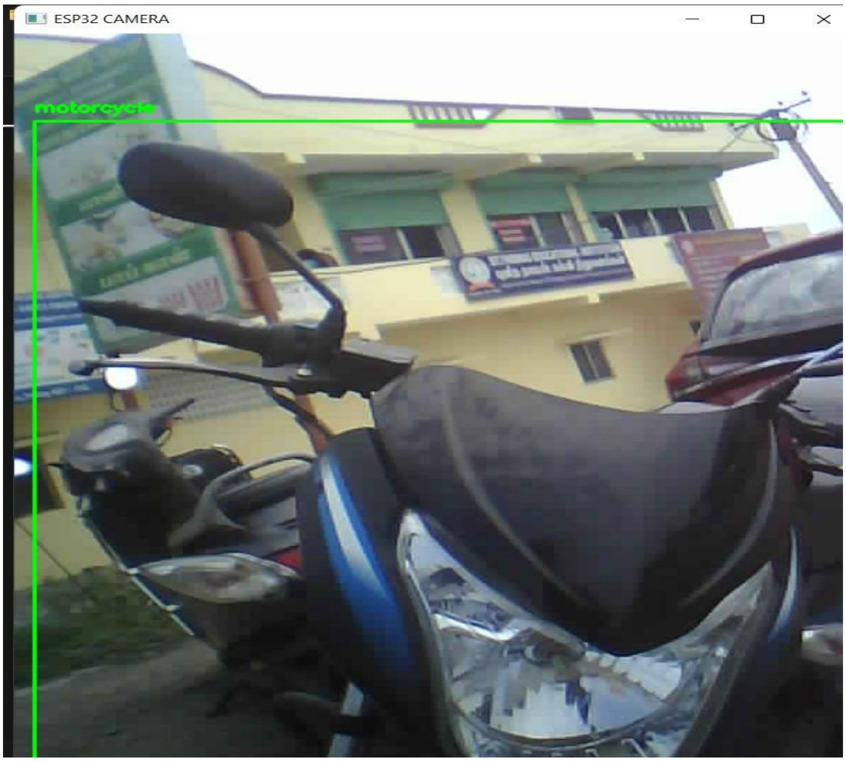

Fig. B2 Motorcycle Detection

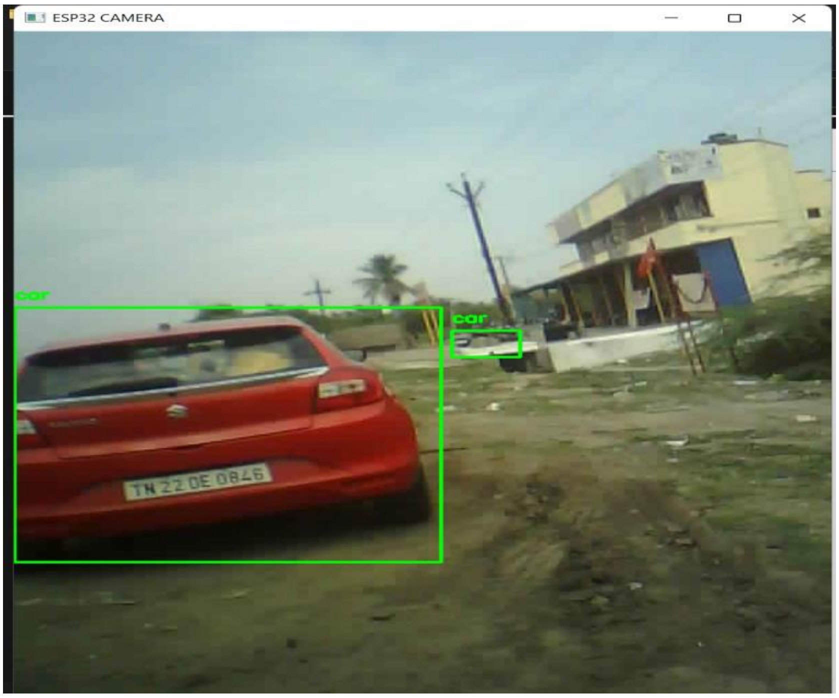

Fig. B3 Motorcycle Detection

Fig. B4 Multi Vehicle Detection

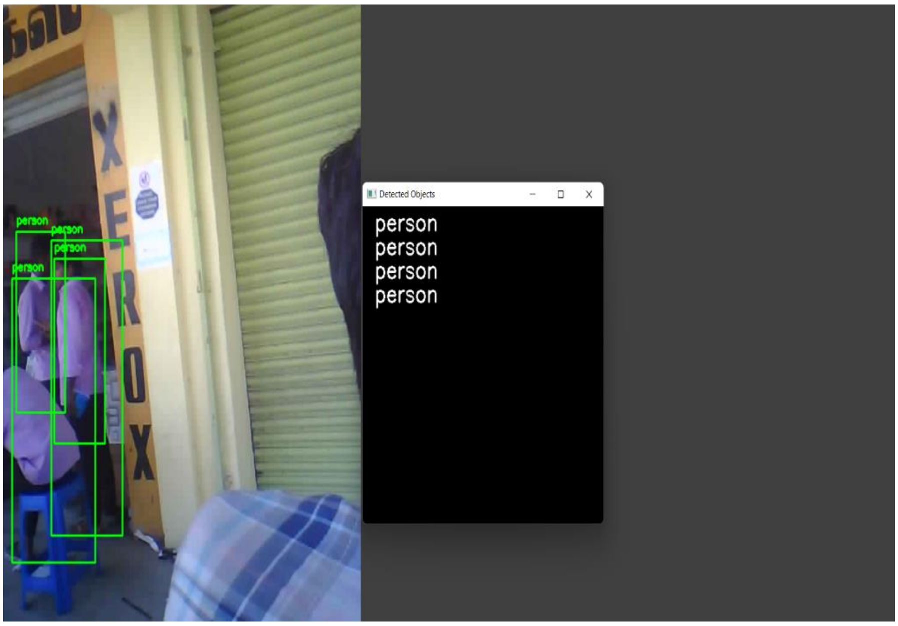

Fig. B5 Multi Person Detection

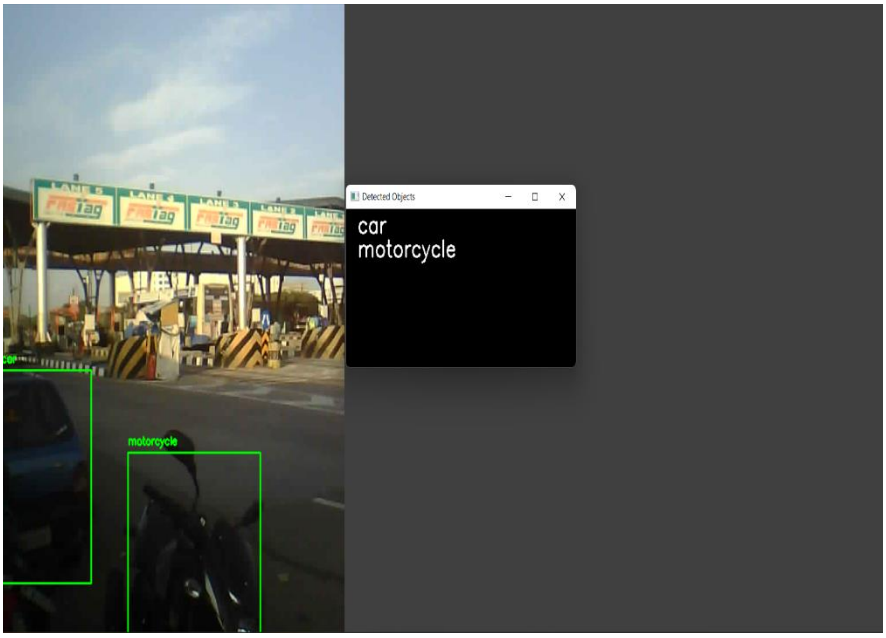

Fig. B6 Vehicle Detection Display

REFERENCES

[1] 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).

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NeurIPS).

[3] Bradski, G. (2000). The OpenCV Library. Dr. Dobb's Journal of Software Tools.

[4] Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

[5] Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering, 82(1), 35-45.

[6] Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., & Savarese, S. (2016). Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[7] Akyildiz, I. F., Pompili, D., & Melodia, T. (2005). Underwater acoustic sensor networks: research challenges. Ad Hoc Networks, 3(3), 257-279.

[8] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences.

[9] Liu, J. W. S. (2000). Real-Time Systems. Prentice Hall.

[10] Buttazzo, G. C. (2011). Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications. Springer Science & Business Media.

[11] Wang, J., Rakotonirainy, A., & Ma, L. (2010). Advanced driver assistance systems: Past, present and future. IFAC Proceedings Volumes, 43(7), 311-318.

[12] Oliver, N., & Pentland, A. (2000). Graphical models for driver behavior recognition in a smartcar. In Proceedings of the IEEE Intelligent Vehicles Symposium.

[13] Roman, R., Zhou, J., & Lopez, J. (2011). On the security and privacy of wireless sensor networks. Computer Networks, 53(12), 2027-2041.

[14] Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A review on internet of things (IoT). International Journal of Computer Applications, 113(1).