Autonomous Car with Collision Detection and Avoidance

Bojja Dheeraj Chowdary
Artificial Intelligence(Medical engineering)
Amrita Viswa Vidyapeetham
Coimbatore, India
cb.ai.u4aim24109@cb.students.amrita.edu

Kumpatla Sai Charan
Artificial Intelligence(Medical Engineering
Amrita Viswa Vidyapeetham
Coimbatore, India
cb.ai.u4aim24124@cb.students.amrita.edu

Lakkireddy Prem Siva Sai Kumar
Artificial Intelligence(Medical Engineering
Amrita Viswa Vidyapeetham
Coimbatore, India
cb.ai.u4aim24125@cb.students.amrita.edu

Pippalla Chirudeep
Artificial Intelligence(Medical Engineering
Amrita Viswa Vidyapeetham
Coimbatore, India
cb.ai.u4aim24137@cb.students.amrita.edu

***Abstract*—The main goal of this project is to develop and design an autonomous vehicle that uses computer vision and embedded systems to prevent and detect collisions. The Raspberry Pi serves as the device's master controller, and an ESP32-CAM module is integrated for object recognition and real-time video transmission. The motor driver controls the vehicle's motion based on information from obstacles that the Raspberry Pi processes. Arduino UNO is used to control the motor driver and the motors including ultra sonic sensors. The ESP32-CAM takes in the surroundings, and the Raspberry Pi recognizes obstacles from the photos using a simple machine learning model or image processing application. The car will automatically stop or change course to avoid a collision once it detects an object at a critical distance. The system can operate safely in dynamic environments and is completely automated. The feasibility of combining inexpensive hardware with AI vision systems for safe, autonomous navigation in small-scale robotic vehicles is demonstrated by this project.**

***Index Terms*—ESP32 Cam Module, Raspberry Pi, Motor driver, Object recognition**

I. Introduction

Autonomous cars are self-driving vehicles capable of navigating without human intervention using sensors and control systems. Even at small scales, autonomous cars are becoming more feasible and accessible due to the rapid advancement of artificial intelligence and embedded systems. With the help of widely accessible hardware parts like the Raspberry Pi, ESP32-CAM module, and motor driver, Ardino UNO. This project aims to create an affordable autonomous vehicle with collision detection and avoidance.

The ultimate goal is to steer the vehicle safely in its environment and move it automatically without human input. The Raspberry Pi uses computer vision techniques to process real-time video captured by the ESP32-CAM module in order to identify obstacles. This states that in order to prevent accidents, the driver controls the vehicle's movement by stopping or steering off. This project demonstrates how a working autonomous navigation system can be constructed using inexpensive microcontrollers and simple image processing. It offers a solid basis for upcoming advancements in AI-driven automation, smart transportation, and robotics. The main components used here are motor driver, ESP32 cam module, raspberry pi, Arduino UNO, jumper wires which are further explained how are they used in the project.

PROBLEMS FACED: 1) Detecting and accurately identifying dynamic obstacles (e.g., vehicles) in real-time is complex, especially in varying traffic and environmental conditions. 2) Autonomous vehicles require advanced algorithms that not only detect obstacles but also predict potential collisions and make quick, accurate driving decisions. 3) Making collision detection systems both fast and accurate is a big challenge in self-driving cars that move quickly.

II. LITERATURE REVIEW

A. Human Detection and Avoidance Control Systems of an Autonomous Vehicle

Authors: T N Nizar

This paper was published in 2020. This paper mainly focuses on short range navigation system of autonomous vechiles. This mainly focuses on detecting the humans. The algorithm used in this paper is YOLO and Fuzzy algorithms. They acquired an average accuracy of 74.4 percent. The best accuracy is found between 2 to 7 meters.

B. Vehicle Detection Techniques for Collision Avoidance Systems

Authors: Amir Mukhtar, Likun Xia, and Tong Boon Tang

This paper reviews the vision based vechile detection. This paper reveals the challenges faced by the system that include vechile shapes and complex environments. The algorithms used in this paper are SVM and HOG. The result given is that no single detection is perfect nad concluded that machine learning methods are most effective for vechile classification.

C. Collision Avoidance in Dynamic Environments Applied to Autonomous Vehicle Guidance on the Motorway
Authors: D. Reichardt, J. Schick

This paper presents a method for autonomous vehicle guidance on motorways. They used multiple vision sensorsto create a real time model of the environment. The algorithms used are Potential feild algorithm and Risk map representation. This does not provide the accuracy of the model but mention that this model is succesful.

D. Data Science Using Warning Systems and Vehicle Crash Detection
Authors: N.Shilpa, K.Veera Kishore,S.Anitha

This paper provides car crash detection with both machine learning and deep learning methods. The algorithms used here are OpenCV and YOLO, additionally they used Logistic regression and Random forest. The accuracy reached 90 percent and also concluded that the model print accident occured if and only if the accident accuracy is more than 90 percent.

E. Autonomous car using CNN deep learning algorithm
Authors: Ilvico Sonata, Lukas

This papers use the Convolutional Neural networks to enable the autonomous car navigation by recognizing the environment. They state that no expensive sensors like LiDar or GPs used, only a camera as the sensor input. They do not provide the accuracy but stated that it is able to drive autonomously without any errors.

III. COMPONENTS

Components are the main part of any model we are going to do. In this Autonomous car the components are Raspberry Pi, Arduino UNO, Motor driver, ESP 32 Cam module, Jumper wires, Motors, Ultrasonic sensors, Servo motors.

A. Microcontroller

Raspberry Pi: This is the main part of the autonomous car. It is like brain of the car that process the sensor data and make decisions for navigation.

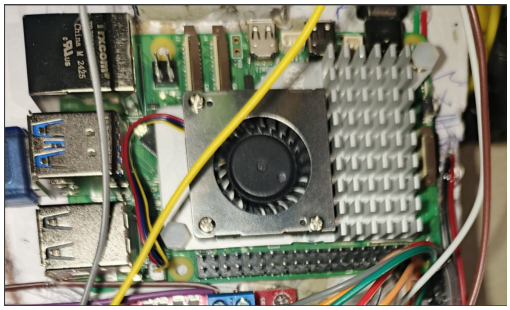

Fig. 1. Raspberry pi 5

The Raspberry Pi 5 model, which has a 2.4 Gigahertz 64-bit quad-core Arm Cortex-A76 processor, is the processor used in the suggested system. The Raspberry Pi 5 has a 40-pin general purpose Input/Output (GPIO) header and 4 or 8 gigabytes of LPDDR4X Random Access Memory. Two USB 3.0 ports, two USB 2.0 ports, and a USB-C power input port that requires 5V at 5A are all included. It is used as the main central processor in the suggested system to manage various inputs, coordinate all activities, and carry out the necessary control actions.

B. Sensors

Ultra Sonic sensors: The main role of this sensors to detect the nearby objects by measuring the sound wave reflections. The main application of this ultra sonic sensor is short-range collision avoidance.

Fig. 2. Ultrasonic sensor

C. Motor Control System

1) Servo Motors: The main role of this is to control the wheel movement of the car. This provides the propulsion and steering of the car.

2) Motor Driver: The model used is L293D. The role of this in car is to provide the power flow to the motors. This adjusts speed and direction based on the micro-controller's commands. Small DC motors and stepper motors can be controlled with

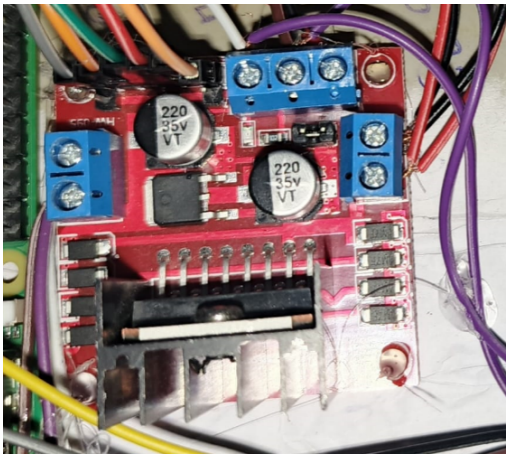

Fig. 3. L293D motor driver

the L293D motor driver integrated circuit, which has sixteen leads. With separate control inputs, the dual full-bridge driver integrated circuit L293D can drive two motors at once. It can manage motor supply voltages of up to 36 volts, with 4.5V as the basic minimum supply voltage. It is appropriate for low-power motor driving applications due to its 600 milliampere

(mA) maximum output current per channel and integrated diodes for back EMF protection.

D. Communication Module

1) ESP32 Cam Module: ESP32-CAM captures real-time images and sends data to the micro-controller. The ESP32

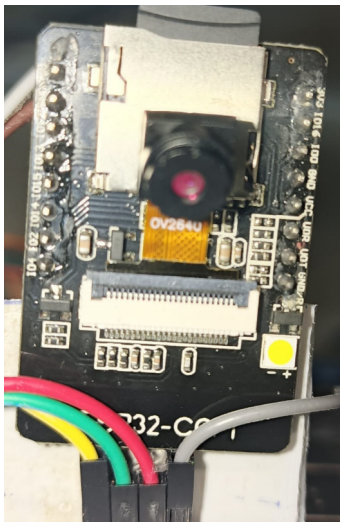

Fig. 4. ESP32 Cam Module

microcontroller processes the images of the given data that are captured by the ESP32-CAM module, which also streams the feed in front of the car in real time. It has a 2-megapixel camera sensor (OV2640). There is no need for external ribbon cables to connect the camera because it is integrated directly into the ESP32 board. Depending on the resolution and processing load, the ESP32-CAM can stream video at up to 1600×1200 (UXGA) resolutions and typically runs at 15 to 30 frames per second. Because of its small size and wireless connectivity, it is widely used in embedded vision and Internet of Things applications.

2) Arduino UNO: Used to control the motor driver, motors and also the ultrasonic sensors.

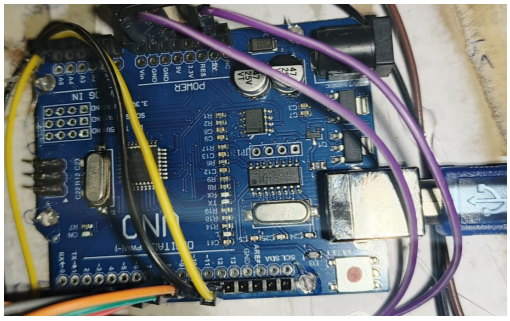

Fig. 5. Arduino UNO

Arduino UNO is an open source microchip board based on 8-bit ATmega-328P microcontroller, it consist of components like Crystal oscillators, Serial communication and Voltage

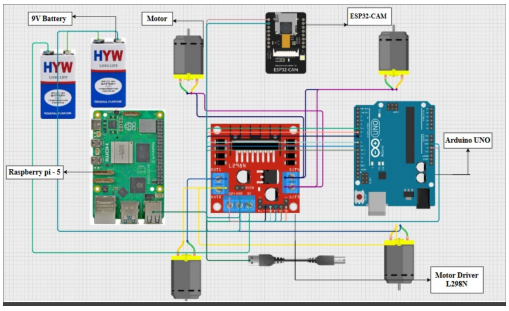

Fig. 6. Circuit diagram

regulators to aid the micro-controller. Arduino Uno has fourteen digital ip or op pins that may be interfaced to different expansion boards (shields) and other circuits, of which six can be operated as Pulse Width Modulation outputs, six Analog ip pins, a Universal Serial Bus connection, an influence barrel jack, an In Circuit Serial Programming header and an adjust switch

E. Some commonly used components:

1) Jumper Wires: These mainly consist of male to male, male-femalle, female-female wires. These are mainly used to connect the main components to each other.

2) Cardboard: This is used to make car base where all the components are placed

3) Steel rod: There is no specific that steel rod should be used but any stiff rod that ultra sonic sensor is attached.

IV. Methodology

A high-speed object detection model called Faster R-CNN (Faster Region-Convolutional Neural Network) was proposed in 2015. It merges detection and localization into one architecture. While the RPN uses CNNs to produce high-quality region proposals, bulky components in the network improve speed and accuracy of detection.

A. Region Proposal Network (RPN)

RPN which reduces proposal time for each image to 10 ms from 2 seconds and improves feature representation by sharing layers with detection stages. It is an essential component of Faster R-CNN. It is responsible for generating possible regions of interest (region proposals) in images that may contain objects. It uses the concept of attention mechanism in neural networks that instruct the subsequent Fast R-CNN detector where to look for objects in the image. The key components of the Region Proposal Network.

a) Anchors boxes: Anchors are used to generate region proposals in the Faster R-CNN model. It uses a set of predefined anchor boxes with various scales and aspect ratios. These anchor boxes are placed at different positions on the feature maps. The parameters are scale and aspect ratio.

b) Sliding Window approach: The RPN operates as a sliding window mechanism over the feature map obtained from the CNN backbone. It uses a small convolutional network (typically a 3×3 convolutional layer) to process the features within the receptive field of the sliding window. This convolutional operation produces scores indicating the likelihood of an object's presence and regression values for adjusting the anchor boxes.

c) objectness score: The objectness score indicates the likelihood that an anchor box contains a meaningful object rather than background.The RPN predicts this score to classify anchors as positive (object) or negative (background) during training.

d) IoU (Intersection over Union): Intersection over Union (IoU) is a metric used to measure the degree of overlap between two bounding boxes. It calculates the ratio of the area of overlap between the two boxes to the area of their union. Mathematically, it is represented as:

$$\text{IoU} = \text{Area of Intersection} \frac{}{\text{Area of Union}}$$

e) Non-Maximum Suppression(NMS): Is used to remove redundancy and select the most accurate proposals, based on the objectness scores of overlapping proposals and keeps only the proposal with the highest score while suppressing the others.

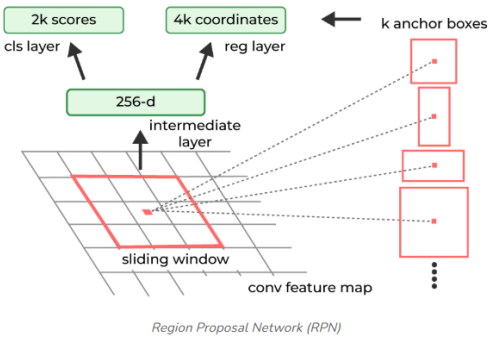

Fig. 7. RPN

B. Region of Interest (RoI) Pooling

The first step is to take the region proposals suggested by the RPN and apply RoI pooling. Region of Interest pooling is used to transform the RPN's variable-sized region proposals into fixed-size feature maps that may be fed into the network's subsequent layers. RoI pooling divides each region proposal into a grid of equal-sized cells then applying max pooling within each cell. This procedure generates a fixed-size feature map for each region proposal, which can be further processed by the network.

C. Feature Extraction

The RoI-pooled feature maps are fed into the CNN backbone resnet 50 (the same one used in the RPN for feature extraction) to extract meaningful features that capture object-specific information. It draws hierarchical features from region proposals. These features retain spatial information while abstracting away low-level details, allowing the network to understand the proposed regions' content.

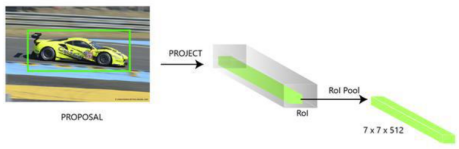

Fig. 8. Region of Interest Pooling

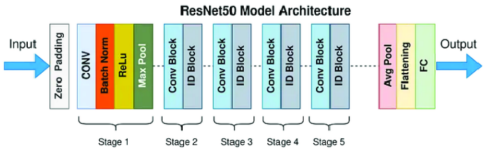

Fig. 9. RESNET 50 Architecture

D. Fully Connected Layers

The RoI-pooled and feature-extracted regions then pass through a series of fully connected layers. These layers are responsible for object classification and bounding box regression tasks.

1) Object Classification: The network predicts class probabilities for each region proposal, indicating the possibility that the proposal contains an object of a specific class. The classification is carried out by combining the features retrieved from the region proposal with the shared weights of the CNN backbone.

2) Bounding Box Regression: In addition to class probabilities, The network predicts bounding box adjustments for each region proposal. The first layer is a softmax layer of N+1 output parameters (N is the number of class labels and background) that predicts the objects in the region proposal. The second layer is a bounding box regression layer that has 4* N output parameters. This layer regresses the bounding box location of the object in the image.

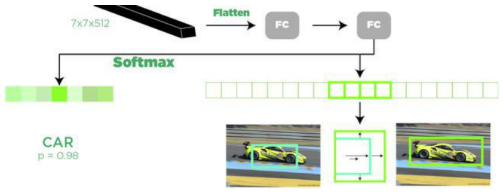

Fig. 10. Bounding Box Regression

3) Multi-task Loss Function: A multi-task loss function that combines classification and regression losses is used by the Fast R-CNN detector. The classification loss computes the difference between expected and true class probabilities. The

regression loss computes the difference between expected and actual bounding box adjustments.

$$(p_i, t_i, v_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, v_i) \quad (1)$$

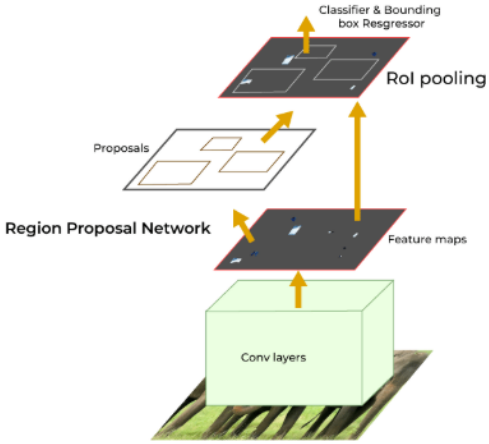

Fig. 11. Faster R-CNN Architecture

E. A* Algorithm

The integrated implementation of an A* (A-star) search algorithm for dynamic path planning of an autonomous vehicle with real-time object detection using FasterR-CNN. ESP32-CAM uses an ESP32-CAM to see its surroundings and in real time plans the optimal direction to avoid obstacles. The ESP32-CAM is used for obstacle detection, and the car avoids them in real time by calculating the directions to move.

1) Object Detection and Obstacle Mapping: The ESP32-CAM module captures video frames which are processed using a TensorFlow-based Faster R-CNN model. The model detects objects such as pedestrians, vehicles, and traffic signs. For each detected object, a bounding box and class label are obtained. The center coordinates of each bounding box are mapped onto a 2D grid representation of the environment, discretized into a 10×10 matrix. Each grid cell is either marked as free (value 0) or as an obstacle (value 1) based on the presence of detected objects.

2) Grid Construction: To enable path planning, the continuous environment is abstracted into a uniform 10×10 grid. Each detected obstacle is projected onto this grid using the formula:

$$gx = cx * (GRIDSIZE/framewidth) \quad (2)$$

$$gy = cy * (GRIDSIZE/frameheight) \quad (3)$$

where (cx, cy) represents the center pixel of the bounding box in the image frame, and (gx, gy) is the corresponding grid coordinate.

3) A Path Planning:* It uses the A* search algorithm to compute for the shortest collision-free path from a predefined start position (usually at the grid center) to a goal location (usually forward in the frame).A uses both cost from the start node and heuristic estimated cost from the goal. the cost function as:

$$f(n) = g(n) + h(n) \quad (4)$$

where g(n) is the cost of the path from the start node to node n, and h(n) is the Manhattan distance from node nto the goal. The algorithm constructs a priority queue and explores neighboring grid cells iteratively in fourdirections (up, down, left, right) but pulls only those that have not been flagged as obstacles.

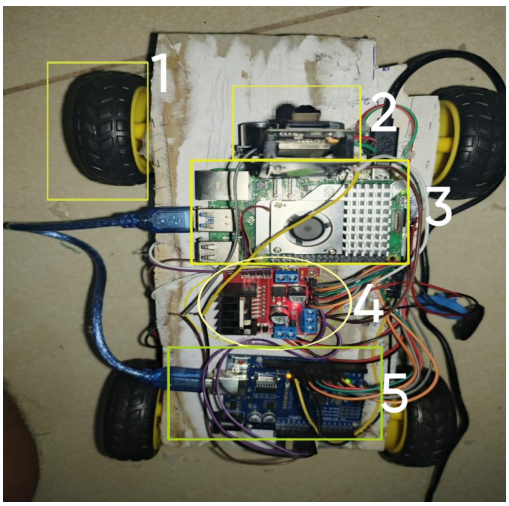

Fig. 12. Autonomous car with motors covered by the wheels labelled as no.1, ESP32 cam module labelled as no. 2, Raspberry pi labelled as no. 3, Motor driver(L293D) labelled as no. 4, and Arduino UNO labelled as no. 5

The above one is the full picture of the model either the autonomous car with all the components labelled.

V. RESULTS

Fig. 13. detection of the objects

Figure 13 is the output received when the object detected by the ESP32 cam module and send the data to the Raspberry pi. The Raspberry Pi detect the objects as you can see in the above picture and command to move left, right etc and send the command to Arduino UNO. The number either co-ordinates in the black box are the grid co-ordinates received after processing by the A* algorithm.

Fig. 14. Detection of objects

The picture in the Fig 14 is also similar picture to that as Fig 12 where in Fig 13 it is the labelled picture of detecting objects on the road of vechiles, trees etc.

The below picture are the box co-ordinates for the Fig 14.

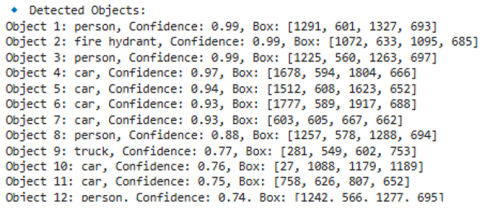

Fig. 15. Box co-ordinates

REFERENCES

[1] T N Nizar, Human Detection and Avoidance Control Systems of an Autonomous Vehicle , 2020.

[2] Amir Mukhtar, Likun Xia, and Tong Boon Tang, Vehicle Detection Techniques for Collision Avoidance Systems, 2015.

[3] D. Reichardt, J. Schick, Collision Avoidance in Dynamic Environments Applied to Autonomous Vehicle Guidance on the Motorway, 2002.

[4] N.Shilpa, K.Veera Kishore,S.Anitha, Data Science Using Warning Systems and Vehicle Crash Detection, 2022.

[5] Ilvico Sonata, Lukas, Autonomous car using CNN deep learning algorithm, 2021.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.