2024-12

Cost Effective IoT-Based Smart System for Avoidance of Obstacle and Intruder Detection

Amrit, Imam Tajnoor Hossain

IUB

Cost Effective IoT-Based Smart System for Avoidance of Obstacle and Intruder Detection

By

Imam Tajnoor Hossain Amrit, Md. Shadman Sakib Talukdar, Mahiyan Rahman Takbir

Summer, 2024

Supervisor:

Mohammad Noor Nabi

Senior Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

December 4, 2024

Dissertation submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

We understand the nature of plagiarism, and we are aware of the university‘s strict policy on the matter. We certify that this is an original work by us. It has been completed under the guidance of Mohammad Noor Nabi. However, others’ written work or software used in any part of this project is properly cited following internationally accepted academic guidelines.

Signature Date

Mahiyan Rahman Takbir,

Name

Signature Date

Imam Tajnoor Hossain Amrit

Name

Signature Date

Md. Shadman Sakib Talukdar

Name

Acknowledgement

First and foremost, we give Allah, the Almighty, praise and gratitude for his blessings throughout our senior project and for enabling us to successfully accomplish it. We appreciate Independent University, Bangladesh (IUB) for providing us with this opportunity. From the bottom of our hearts, we would want to thank everyone who helped us with the knowledge needed to finish this project. We would like to express our profound gratitude to Mohammad Noor Nabi, a senior lecturer at Independent University in Bangladesh, who served as our supervisor and mentor and whose assistance, guidance, and recommendations were very helpful in completing our project. He guided and helped us, and our project advanced effectively. Additionally, we would like to extend our sincere gratitude to the committee members for their thoughtful remarks and suggestions, as well as for allowing our defense to be a fun experience.

Letter of Transmittal

4 December, 2024
Mohammad Noor Nabi,
Senior Lecturer of Independent University, Bangladesh
Aftabuddin Ahmed Road, Block-B, Bashundhara R/A Dhaka

Subject: Letter of Submission of Senior Project Report, Fall 2024.

Dear Sir,
With humble submission and due respect, we are writing this letter to inform you that we have completed our Senior Project and its report. We did our best as we could with high passion, dedication and enthusiasm. We hope that you would be kind enough to consider and forgive any mistakes in preparing this report and oblige thereby.

Sincerely,
Mahiyan Rahman Takbir
Md. Shadman Sakib Talukdar
Imam Tajnoor Hossain Amrit

Evaluation Committee

. . . .
Signature

. . . .
Name

. . . .
Supervisor

. . . .
Signature

. . . .
Name

. . . .
Internal Examiner

. . . .
Signature

. . . .
Name

. . . .
External Examiner

. . . .
Signature

. . . .
Name

. . . .
Convener

Abstract

This study introduces a cost-effective integrated system that significantly enhances the safety and security of autonomous vehicles by merging obstacle avoidance and intrusion detection functionalities through Internet of Things (IoT) technologies. The system employs ultrasonic sensors for accurate obstacle detection, with an Arduino Uno and Raspberry Pi 3 handling data processing and control tasks. Facial recognition technology is incorporated to monitor and identify unauthorized individuals in real time. This integration allows the vehicle to autonomously navigate while also securing itself against potential intruders, all achieved with readily available and affordable components. The system demonstrates strong performance in detecting and avoiding obstacles in straightforward scenarios, although accuracy may diminish in more complex environments. The facial recognition component achieves consistent and reliable results under controlled conditions. By combining these technologies in a budget-friendly solution, this research offers practical advancements in intelligent and secure autonomous vehicle systems, suitable for applications in various domains including robotics and smart vehicle technology.

Keywords— IoT; Facial Recognition; Autonomous; Intrusion Detection; Obstacle Avoidance; Low Cost;

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Significant progress has been made in the development of vehicular autonomy in recent years, with an emphasis on lowering driving-related human error and raising safety. An autonomous vehicle uses a variety of sensors, including radar, lidar, ultrasonic, infrared, and cameras, to sense its surroundings in order to travel to its destinations without the assistance of a human. These cars are made to reduce accidents brought on by human distractions, including using a phone while operating a motor vehicle [1].

Numerous advantages are offered by the autonomous car sector, such as increased fuel efficiency, a decline in accident rates, and lower pollution levels. These cars offer a safer option to human drivers in practically all circumstances since they can function securely even if the driver is tired [1]. Nevertheless, there are still issues that need to be resolved, like figuring out who is at fault in an accident and having autonomous systems have trouble identifying roadway signs and human behavior [1].

Vehicle security systems have developed in tandem with autonomous navigation to thwart theft and unwanted access. Low security levels result from traditional alarm systems' inability to differentiate between a car owner and a burglar. These technologies improve security and prevent unwanted entry by utilizing face recognition technology to precisely identify the automobile's owner [2]. Widely used biometric technology, facial recognition compares a person's face traits to a database that is recorded, allowing for precise recognition [2].

The choice of sensor is essential in obstacle avoidance to guarantee dependable performance. Numerous sensors, including infrared, ultrasonic, and laser scanners, have been used in a variety of applications to identify and steer clear of obstructions. A robot using optical sensors, for example, would have trouble in areas with glass walls, highlighting the significance of selecting the appropriate sensors depending on the particulars of the barriers [3]. Particularly well-liked for their affordability and ease of design, ultrasonic sensors are perfect for low-fidelity prototypes and small-scale robotics [4]. The area of autonomous automobiles and robotic systems has seen a revolutionary change because of the integration of several Internet of Things components, including sonar sensors, HD cameras, ESP32, Arduino Uno, and Raspberry Pi 3. The open-source Arduino Uno microcontroller is well known for its user-friendliness and strong community support. Due to its easy programming and flexible input/output features, it is widely used in educational projects and prototyping. This project uses an Arduino Uno to communicate with the sonar sensor and an L298N motor driver to operate the DC motors [4].

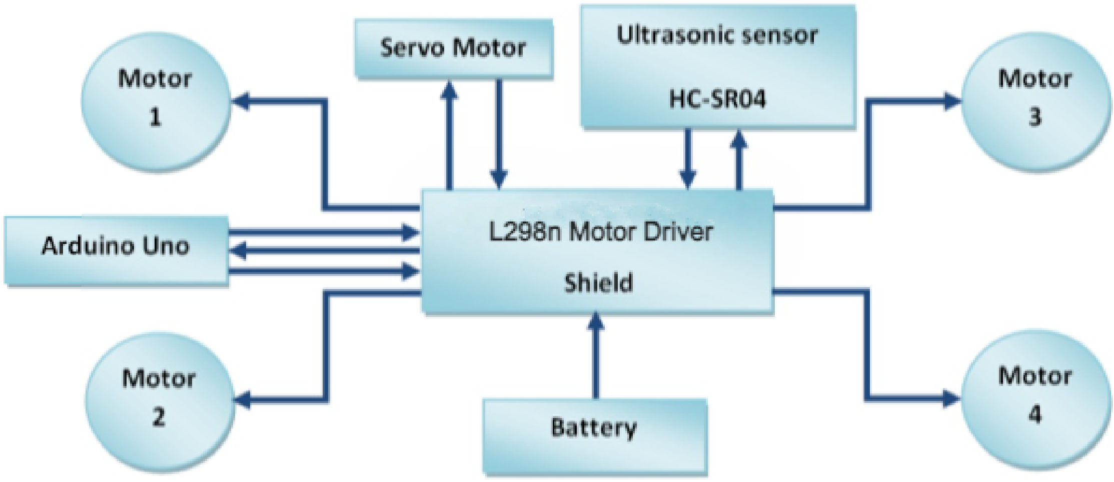


Figure 1.1: Obstacle Avoidance using Arduino

The system's central processing unit is the Raspberry Pi 3, a small and potent single-board computer. With its quad-core processor, large RAM capacity, and variety of networking choices, the latest version of the Raspberry Pi is an excellent choice for managing intricate activities like data analysis and picture processing. This project uses facial detection to distinguish between known and unknown people by processing the HD camera's collected photos [5]. The basic steps for a facial recognition system is depicted in Fig. 1.2.

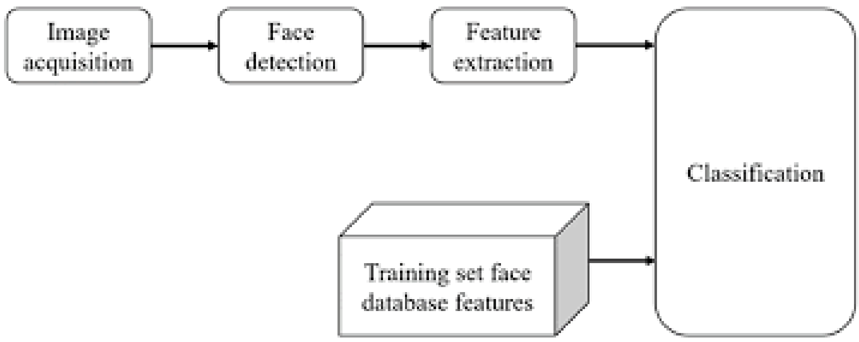


Figure 1.2: Facial Recognition Basic Steps

An ultrasonic sensor called a sonar sensor is necessary for spotting and avoiding obstacles. By generating ultrasonic waves and timing how long it requires for the waves to return, it calculates the distance to objects. The Arduino Uno processes this data to calculate the closeness of obstacles and adjust the vehicle's path accordingly. Sonar sensors are valued for their precision, dependability, and cost-effectiveness, making them an increasingly common option in numerous applications for robotics [3, 4]. This paper describes the concept and implementation process of an autonomous unmanned vehicle that makes use of Internet of Things devices to combine intrusion detection and obstacle avoidance capabilities. A Raspberry Pi 3, an HD camera, DC motors, an Arduino Uno, a sonar sensor, and 12V Li-ion battery are all used in the system. The car gives notifications when it detects impediments, recognizes people it doesn't know, and navigates on its own. A comprehensive approach for navigational autonomy and real-time security surveillance is provided by this integration of diverse IoT components. The project's autonomous car shows how cutting-edge sensors and Internet of Things gadgets may be used to produce dependable obstacle evasion and intrusion detection. The system can navigate complex surroundings, prevent collisions, and improve security by conducting real-time monitoring and alarms by utilizing the advantages of each component. With its insights and solutions for creating more complex and secure autonomous systems, this research advances the fields of smart robotics and IoT.

1.2 Objectives

The primary objective of this research is to develop a comprehensive system for obstacle avoidance and intrusion detection using IoT devices. This system aims to enhance the maneuverability and security of autonomous robotic vehicles, leveraging various sensors and microcontrollers. The specific objectives of the project are as follows:

Develop an Autonomous Navigation System: Create a robotic vehicle capable of navigating autonomously in an unknown environment by detecting and avoiding obstacles in real-time.

Integrate Advanced Sensor Technologies: Utilize a sonar sensor to detect obstacles accurately and avoid collisions.

Leverage IoT Components for Enhanced Functionality: Employ Arduino Uno and Raspberry Pi 3 microcontrollers for processing sensor data and controlling the robotic vehicle's movements.

Implement Facial Recognition for Security: Integrate facial recognition technology to identify and differentiate between authorized users and potential intruders. Use an HD camera in conjunction with the Raspberry Pi 3 to capture images and perform facial recognition, ensuring that the vehicle can respond appropriately to unauthorized access attempts

Design a Robust Alert System: Develop a system that sends real-time notifications, enhancing the overall security and monitoring capabilities.

Achieve Cost-Effective and Reliable Performance: Focus on creating a cost-effective solution that uses readily available components while maintaining high reliability and performance. Test and evaluate the system in various environmental conditions to ensure robustness and effectiveness in real-world scenarios

Promote Future Research and Development: Provide insights and data that can be used for future enhancements and research in the fields of autonomous navigation and IoT-based security systems. Highlight the potential applications of the developed system in various industries, such as smart homes, autonomous vehicles, and robotics. By achieving these objectives, the project aims to contribute to the advancement of autonomous systems and IoT technologies, offering practical solutions for obstacle avoidance and intrusion detection.

1.3 Scopes

The project on Obstacle Avoidance and Intrusion Detection Using IoT Devices has the potential for significant impact in several fields. The scope of this project can be broadly categorized into various domains where its application, further development, and scalability are feasible:

Autonomous Navigation Systems The system demonstrated effective obstacle detection and avoidance capabilities, albeit with some limitations. As a result, its scope can extend to:

- Autonomous vehicles (e.g., drones, robots) used in various industries such as agriculture, logistics, and security patrols.
- Smart wheelchairs for the disabled, where safe and reliable obstacle avoidance is paramount.
- Warehouse automation systems, where autonomous robots could navigate complex environments while detecting obstacles and avoiding collisions.

Security and Surveillance With the integration of face recognition, the project opens doors to advanced security solutions. Some applications include:

- Intrusion detection systems that not only identify unknown faces but also flag suspicious individuals.
- Smart door locks that integrate facial recognition for access control, ensuring only authorized personnel can enter specific premises.
- Surveillance drones equipped with obstacle avoidance and face detection features to monitor sensitive areas such as borders, military zones, and large public spaces.

IoT and Smart Homes As the project is based on IoT technology, it has potential applications in the rapidly growing field of smart home systems:

- Home security systems that use face recognition to allow or restrict entry, while the vehicle's obstacle avoidance mechanism could be adapted to monitor and control home environments.
- Remote monitoring and control, where homeowners can track intruders and receive alerts for potential security breaches.

Robotics and Automation The integration of Raspberry Pi, Arduino, and sonar sensors in this project offers a strong base for robotics development. Future work in this area could lead to:

- Autonomous delivery robots in urban environments that can navigate sidewalks, avoid pedestrians, and recognize delivery points.

- Robotic assistants in hospitality or healthcare settings that move around autonomously while avoiding obstacles and identifying individuals.

IoT-Based Surveillance in Public Spaces The face recognition system integrated with IoT capabilities provides scope for public space monitoring:

- Crowd management at large events such as concerts, sports events, or airports, where the system could identify individuals in real time.
- Smart cities infrastructure, where interconnected IoT devices manage security systems and monitor traffic or public safety concerns.

Military and Defense Applications The system's capabilities could also be adapted for use in military and defense applications:

- Unmanned ground vehicles for reconnaissance missions, which can navigate challenging terrain while identifying threats or unknown individuals.
- Border security systems that use face recognition and obstacle detection to enhance surveillance and intruder detection in remote or difficult-to-reach locations.

1.4 Problem Statement

An automated system with actual time obstacle avoidance and unknown person recognition capabilities is vital in contexts where human involvement is uncertain and obstructions present navigational risks. Conventional surveillance systems frequently lack the intelligence and adaptability needed to react quickly to changing circumstances, which can result in security breaches and possible safety risks. Furthermore, the accessibility and scalability of current systems may be constrained by their reliance on expensive or complicated configurations.

This endeavor intends to design an integrated hardware and software solution that makes use of commonly accessible parts and microcontrollers to produce a reliable obstacle avoidance and person detection system in order to handle these challenges. Through the utilization of technologies including sonar sensing, camera imaging, and microcontroller-based processing, the suggested system aims to furnish users with prompt alerts and notifications, permitting anticipatory reactions to possible hazards. It must be possible for the system to blend in with both indoor and outdoor situations with ease, all the while preserving its dependability, effectiveness, and cost. The key objectives of this project include:

- Designing a hardware architecture comprising sensors, microcontrollers, and actuators to enable obstacle detection and avoidance.
- Implementing intelligent algorithms for real-time analysis of sensor data to distinguish between known and unknown entities.
- Developing a communication protocol for transmitting alerts and notifications to user devices.

The remainder of this report comprises seven more chapters. Chapter 2 reviews the related works and relations with undergraduate courses offered with the study. Chapter 3 discusses about the dataset. The following chapter, chapter 4 outlines the methodology used for the problem. Chapter 5 analyzes the investigated result. Chapter 6 illustrates the impact the study may have on the society. The following chapter, chapter 7 highlights the lesson learned. Finally, the whole work was abstracted in the last chapter.

Chapter 2

Literature Review

2.1 Relationship with UG studies

The university places a high value on teaching, learning, and the learning process in its commitment to the development of responsible, mature, and educated citizens. Our undergraduate education gave us the information and abilities that enabled us to research the subject and create the project. If these classes hadn't been finished before beginning this assignment, it would have been harder. In addition to that, the group and individual projects we completed for our undergraduate classes were really helpful to us with this project. The knowledge and talents gained in undergraduate courses have aided in developing, testing and declaring the originality of this study. It would be more difficult if these courses were not covered while working on this project. Some of these courses are:

An Introduction to Programming: This course teaches and provides the necessary Information about introducing a new programming language and programming fundamentals, including variable declaration, iteration, conditional expressions, switch-case, arrays, and functions.
PHY 101-University Physics I: This course covered the fundamental principles of physics, including mechanics, thermodynamics, and waves. The rigorous study of these topics developed my problem-solving skills and my understanding of physical principles that are essential for understanding the behavior of sensors and the dynamics of robotic movements.
PHY 102-University Physics II: Building on the concepts from PHY 101, this course focused on electromagnetism, optics, and modern physics. The knowledge gained here is directly applicable to understanding how sensors work, especially those involving electromagnetic principles, and how they can be integrated into a robotic system for obstacle detection and intrusion prevention.
CSE 104-Electrical Circuit Analysis: This course provided a comprehensive understanding of electrical circuits, including Ohm's law, Kirchhoff's laws, and the analysis of AC and DC circuits. This knowledge is crucial for designing and troubleshooting the electronic components of the project, ensuring that all sensors and microcontrollers are correctly powered and functioning.
CSE 203 Data Structure: A data structure is a specialized format used to organize, process,

retrieve, and store data. There are various basic and complex data structures, all of which are intended to organize data for a specific purpose. Data structures make it simple for users to obtain and deal with the required data. Data structures organize information so machines and humans can understand it. It is critical to use data structures and select the appropriate data structure for each task. An inappropriate data structure may result in poor run times or unresponsive programming. This course taught how to handle and manipulate complex arrays, objects, classes, an array of objects, objects of the array, nested arrays, nested objects, etc. As coding of a model involves many complex data structures, the knowledge gained from this course makes handling them much easier.

CSE 213 Object-Oriented Programming: The concept of objects underpins object-oriented programming. Object-oriented programming defines data structures or objects, each with its traits or attributes. Each object can also have its own set of processes or methods. Objects that interact with one another are used to create software. OOP can also be utilized in manufacturing and design applications because it reduces the effort required. It can be used to create blueprints and flowcharts, for example. It helped to write reusable functions that were needed in many section of the study.

CSE 216-Microprocessor Interfacing & Assembly Language: Here, we gained hands-on experience with microprocessors, learning how to interface them with various peripherals and write assembly language programs. This knowledge is directly applicable to programming the microcontrollers (Arduino Uno, Raspberry Pi) used in this project, enabling effective communication between the sensors and the control system.

CSE 310-Electronics II: An extension of Electronics I, this course delved deeper into advanced electronic circuits, including analog and digital integrated circuits. The skills acquired here are vital for designing robust and efficient circuits that interface with the various sensors and ensure the system's reliability.

CSE 317: Numerical Methods: The benefit of using a numerical approach is that it can address issues without using an analytical answer. Numerical techniques provide the additional benefit of just evaluating fundamental operations and functions, such as addition, subtraction, multiplication, and division. Numerical analysis is used to examine ways for providing approximations to numerical solutions in cases when finding the true solution rapidly is doubtful, and to try and improve upon these methods in order to lower the amount of error produced by computer calculation. This course helped understand the mathemetics behind all those complex problems and how to find optimal solution. Also, this course taught us MATLAB and PYTHON which are most widely used in real world for this kind of study.

2.2 Related works

Adarsh et. al. compared the effectiveness of infrared (IR) and ultrasonic (US) sensors [6]. A variety of barriers, including paper, cardboard, sponge-like substance, plastic, lumber, rubber, and tile, were used in the comparison. In order to assess each sensor's performance against various materials, the study concentrated on the centimeter-based criterion of "calculated distance versus genuine distance." In a 2017 study, Pavithra et. al. examined the features of several sensors that are employed for avoiding obstacles in both indoor and outdoor settings [7]. They came to the conclusion that because of the latter's range restrictions and implementation complexity, ultrasonic sensors are better suited for obstacle detection than infrared and laser sensors. In their 2015 study, Jaycil and Randy emphasized the benefits and drawbacks of the numerous sensors utilized in the construction of fully autonomous cars [8]. They made the point that multiple sensors are required for complete autonomous vehicle operation because no single sensor can offer all the information needed. In 2011, Domínguez et. al. talked about the advancement of driverless cars, which rely on sensor commands for navigation [9]. They mentioned the usage of sensors, LiDAR technology, radar, GPS, and other equipment, emphasizing the superior data processing powers of ultrasonic and RADAR sensors over human drivers. The development of autonomous electric vehicles through the use of simulation instruments was detailed by Jie et. al. [10]. To create a closed-loop control architecture for autonomous vehicles, they created a simulation framework based on MATLAB/Simulink. By incorporating automotive sensors, road conditions, and impediments, they were able to enhance the control logic to handle local path planning.

A single-beam mechanically inspecting profile sonar for submerged obstacle identification was reported in a study [11]. The cone-shaped beam produced by this sonar is perfect for identifying objects close to the surface. The study's findings were encouraging for sonar use close to the water-air border. Other research combined appearance- and range-based techniques with several sensors to improve obstacle detection [12]. These methods frequently combined cameras and ultrasonic sensors to offer complete identification and measurement capabilities.

Another study divided the variables influencing face recognition system accuracy into two categories: extrinsic variables (partial occlusion, position variance, and illumination) and intrinsic variables (aging and facial expressions) [13]. Authors [14] stated that, a strong facial recognition system should be able to recognize faces in both photos and videos, work in real time, withstand changes in lighting, and be insensitive to changes in gender, hair color, or ethnicity. Faces from every viewpoint should also be detected by the system. Face recognition is accomplished using a variety of methods, including Eigenface, artificial neural networks, the Hidden Markov Model (HMM), as well as Support Vector Machine (SVM). The Eigenface method produced an accuracy of 85% on average [15]. With 91.67% accuracy, a hybrid algorithm that used Eigenface and Haar Cascades identified numerous faces [16]. The average accuracy displayed by neural networks was 96.84% in a study [17].

This literature review synthesizes various approaches and technologies for obstacle detection and avoidance, sensor integration in autonomous systems, and advancements in face recognition, highlighting the progress and challenges in these areas.

Chapter 3

Dataset Description

The dataset utilized for this project comprises 15 images of the authors, captured under varying conditions to test the robustness of the face recognition system. The images are categorized and organized in a structured manner, where each author's images are stored in separate, appropriately named folders. Below is a detailed breakdown of the dataset:

1. Image Collection The images were taken from different angles to simulate realistic conditions:

- Front-facing images to capture the entire face directly.
- Right-side profiles for capturing the facial structure from the right.
- Left-side profiles to account for the left-side facial view.

Variations in lighting conditions were included:

- High brightness images to simulate well-lit environments.
- Low brightness images for dimly lit or poorly illuminated settings.

2. Image Specifications Each image was resized to remain within a maximum resolution of 600x600 pixels. The resizing process ensures that the images maintain clarity while minimizing file size for efficient processing. The resizing standardizes image dimensions, contributing to consistent results during the training and recognition process.

3. Folder Structure Each author has their own dedicated folder, with the folder name corresponding to the author in the images. For example, folder "takbir" contains five images of the author Takbir from various angles and lighting conditions. Other authors, such as Tajnoor and Shadman, have their own folders named after them. This organization allows for clear demarcation of individual datasets, aiding in the training phase where each folder represents a distinct class for recognition.

4. Dataset Composition A total of three authors contributed images to the dataset. Each author has 5 images in their respective folder, totaling 15 images across the dataset.

5. Purpose of the Dataset This dataset is primarily used for training the pre-trained facial recognition model, ensuring it can detect and recognize faces even under variable conditions such as angle changes or lighting variations. The classification of images into specific folders enables the system to distinguish between different individuals accurately. This dataset represents a diverse set of facial characteristics and conditions, making it suitable for evaluating the system's ability to recognize faces reliably.

Chapter 4

Methodology

4.1 List of Tools Used

Several tools were used in this study to achieve the goals. So far, a number of tools and libraries have been made available to guarantee the security and privacy of image processing, AI models and data. The tools that were used in this study are discussed next briefly.

4.1.1 Softwares as tools

- *Google Colab - web IDE:* Google Colaboratory, sometimes known as "Google Colab," is a Google Research product. Machine learning, data analysis, and education are three areas where Colab excels. It makes it possible for anyone to write and run arbitrary Python code through a web browser. Technically speaking, Colab is a hosted Jupyter notebook service that offers free access to computer resources, including GPUs, and requires no setup to use.

- *Python – Language:* A high-level, all-purpose programming language is Python [18]. Its design philosophy, which heavily relies on indentation, prioritizes code readability. The dynamic typing language Python makes use of garbage collection. It supports a variety of programming paradigms, such as functional, object-oriented, and structured programming.

- *OpenCV – Library:* OpenCV [19] (Open Source Computer Vision Library) is an open-source computer vision and machine learning software library. It provides a common infrastructure for computer vision applications and speeds up the use of machine perception in commercial products. It includes support for real-time image processing and object detection, making it widely used in applications such as face recognition, object tracking, and augmented reality.

- *Face Recognition – Library:* The face-recognition [20] library provides a simple, powerful interface for detecting faces and recognizing people in images or videos. Built on top of dlib's deep learning-based face recognition algorithm, it is considered to be one of the easiest libraries for facial recognition, providing highly accurate results even with relatively simple setups.

- *Imutils – Library:* Imutils [21] is a set of convenience functions to make basic image processing functions such as resizing, rotating, and displaying images easier with OpenCV. This library simplifies many routine tasks like adjusting image sizes, computing skeletons, or visualizing outputs in Python.

- *Pickle – Library:* Pickle [22] is a Python standard library module used for serializing and deserializing Python object structures, also known as "pickling" and "unpickling". This is essential for saving model objects, data, or any Python object to disk and retrieving it later, which is particularly useful in machine learning workflows for saving trained models or processing states.

- *Pushbullet:* The Pushbullet notification platform sends messages to Pushbullet, a free service to send information between one's phones, browsers, and friends.

4.1.2 Hardwares as tools

- *Arduino Uno:* Arduino is a microcontroller. It is part of a computer that runs one program again and again. Arduino board uses a variety of microprocessors and controllers [23]. The boards consist of sets of digital and analog input/output (I/O) pins that may be interfaced to various boards or breadboards and other circuits. The boards feature Universal Serial Bus(USB), serial communications interfaces on some models, which are also used for loading programs from personal computers. The integrated development environment (IDE) that are generally used to dump the project. It consists of 54 digital

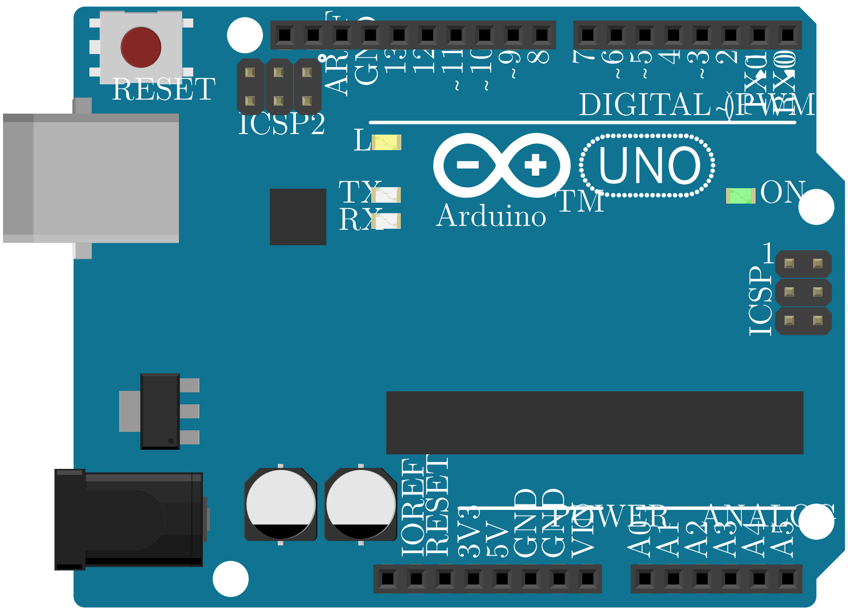


Figure 4.1: Arduino Uno

input/output pins (of which 14 can be used as PWM outputs), 16 pins are analog inputs, 4 UARTs (hardware serial ports), a clock speed of 16 MHz crystal oscillator, a USB connection, ICSP header a power jack, and a reset button. It simply started by connecting to a computer with a USB cable or applying AC or DC power using AC- to- DC adapter or battery. Arduino mega power pins are 5V, 3V, and GND. Pins are supplied less than 5V voltage the board becomes unstable and if the supply voltage is more than 12V there is a chance of damaging of voltage regulator because of overheat.

- *Raspberry Pi:* Similar to that of a simple desktop PC but with less processing capability, the Raspberry Pi is a single-board computer [24]. Keyboard and mouse peripherals can be connected to its USB ports, and an HDMI port allows you to connect a monitor. The Raspberry Pi has been available in multiple iterations throughout the years, each with differing performance levels and intended uses. The ARM architecture, which is the basis for many smartphones, powers the Raspberry Pi's processors. Windows 10, Linux variants, and Microsoft's IoT Core are among the operating systems it can run. The Raspberry Pi may be used to develop applications in addition to controlling devices, unlike the Arduino, which is primarily used to control hardware through GPIO pins. The first 64-bit Raspberry Pi was the most recent model, the Raspberry Pi 3 Model B, which was released in February 2016. Because it has Bluetooth and Wi-Fi built in, it is ideal for applications like motion and face detection.

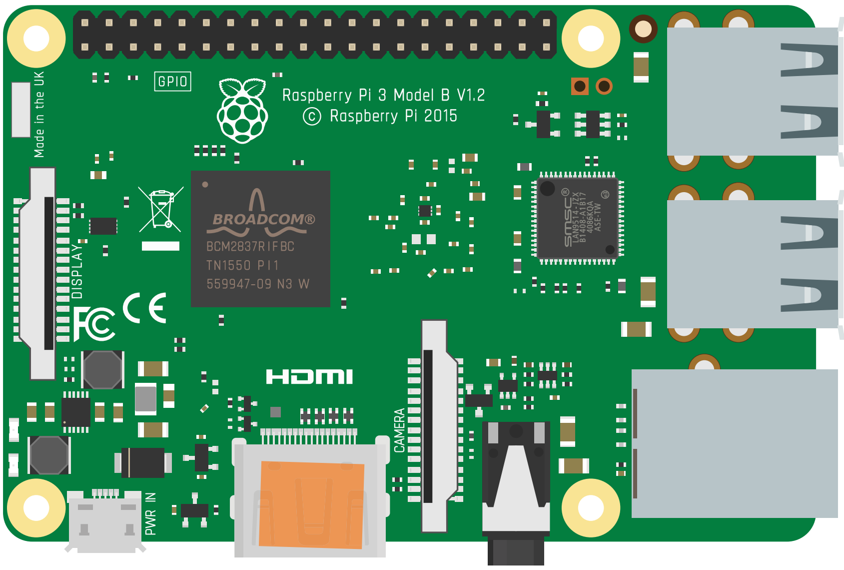


Figure 4.2: Raspberry Pi

- *L298N Motor Driver:* The L298N motor driver is a crucial component used in this project to control the movement of DC motors within the obstacle-avoidance system. This dual H-bridge motor driver IC allows independent control of two DC motors in both forward and reverse directions, making it an ideal choice for robotic applications requiring mobility [25]. In this project, the L298N motor driver is integrated with the Arduino Uno, which receives control signals to drive the motors based on the input from the sonar sensor. The working process begins with obstacle detection: when the sonar sensor detects an obstacle within a predefined range, such as 3 feet, it sends a signal to the Arduino. Upon receiving this signal, the Arduino triggers the L298N to stop the motors, thereby preventing the vehicle from moving forward. Following this, the L298N motor driver manages the vehicle's directional changes. After stopping, the motor driver receives instructions from the Arduino to either reverse or turn the vehicle, allowing it to avoid the obstacle and continue moving once a clear path is detected.

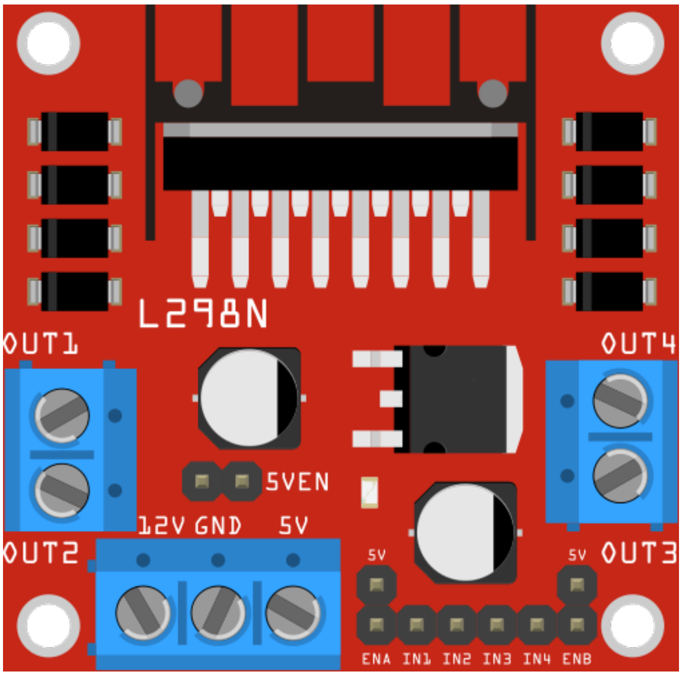


Figure 4.3: L298n Motor Driver

- *HC-SR04 Ultrasonic sensor:* The HC-SR04 ultrasonic sensing element uses echo sounder to work out the gap to an object like dolphins do. It offers wonderful non-contact detection with high accuracy associated stable readings in an easy-to-use package from two cm to four hundred cm. Its operation isn't restricted by daylight or black material. It comes with a transmitter and recipient module. The HC-SR04 Ultrasonic Sensor is a key component used for obstacle detection in this project. This sensor operates by emitting ultrasonic waves from its transmitter, which travel through the air and reflect off objects in their path. The sensor then measures the time it takes for the reflected waves to return to the receiver, calculating the distance between the sensor and the object based on the speed of sound. In the context of this project, the HC-SR04 is connected to the Arduino Uno, continuously monitoring for obstacles in front of the vehicle. When an object is detected within a specified range, such as 3 feet, the sensor sends a signal to the Arduino, prompting it to initiate further actions like stopping the motors or changing direction. Its reliable distance measurement capabilities make it ideal for use in autonomous vehicle navigation, ensuring accurate obstacle detection and avoidance.

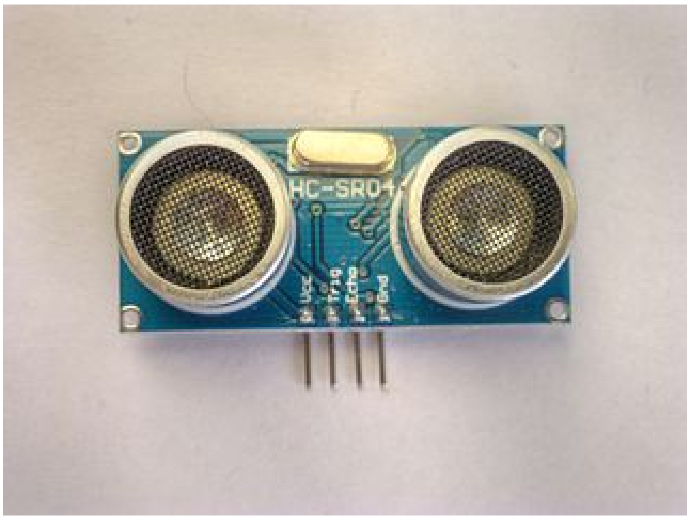


Figure 4.4: HC-SR04 Ultrasonic sensor

4.2 Costing

The total cost breakdown of the hardware equipments are listed below:

Table 4.1: Costing of tools used

Tools Name	Price (in USD)
Arduino Uno	3.35$
Raspberry Pi 3b with 32GB memory card	39.74$
L298n Motor Driver	1.13$
DC Motor with 4 Wheels	4.18$
HC-SR04 Ultrasonic Sensor	0.78$
Plastic Empty Container	0.84$
Jumper Wires	0.50$
Breadboard	0.67$
HD Webcam	11.71$
Total	63$

Costing of the software tools used in this study is free.

4.3 Methods

The methodology for this study is based on integrating various IoT devices and sensors to develop a system capable of obstacle avoidance and intrusion detection. The system is designed using a combination of Arduino, Raspberry Pi, sonar sensors, and an HD camera.

4.3.1 Components Setup

The key components used in the system include:

- *Raspberry Pi 3 Model B:* Acts as the primary processing unit, responsible for facial recognition and data analysis.
- *HD Camera:* Connected to the Raspberry Pi, the camera captures real-time images for facial recognition.
- *Arduino Uno:* Serves as the controller for the obstacle detection mechanism.
- *Sonar Sensor:* Utilized for detecting obstacles within a specified range (approximately 3 feet).
- *DC Motor with Wheel:* Facilitates the movement of the vehicle.
- *L298n Motor Driver:* Controls the motor's direction and speed based on input from the Arduino.
- *12V Li-ion Battery::* Powers the entire system.

The hardware components are assembled to form a small vehicle equipped with the necessary sensors and cameras. The Raspberry Pi is interfaced with the HD camera through the CSI port, while the Arduino is connected to the sonar sensor and motor driver.

4.3.2 System Operation

The system operation is divided into two primary tasks: obstacle avoidance and facial recognition.

Obstacle Detection and Avoidance

- Upon activation, the Arduino and sonar sensor are powered on.
- The sonar sensor continuously emits ultrasonic waves and listens for their reflection to detect the presence of obstacles within its range.
- If an obstacle is detected within 3 feet, the sensor sends a signal to the Arduino, which then commands the motor driver to stop the vehicle.
- The vehicle then changes its direction based on the detected obstacle's position, avoiding potential collisions.

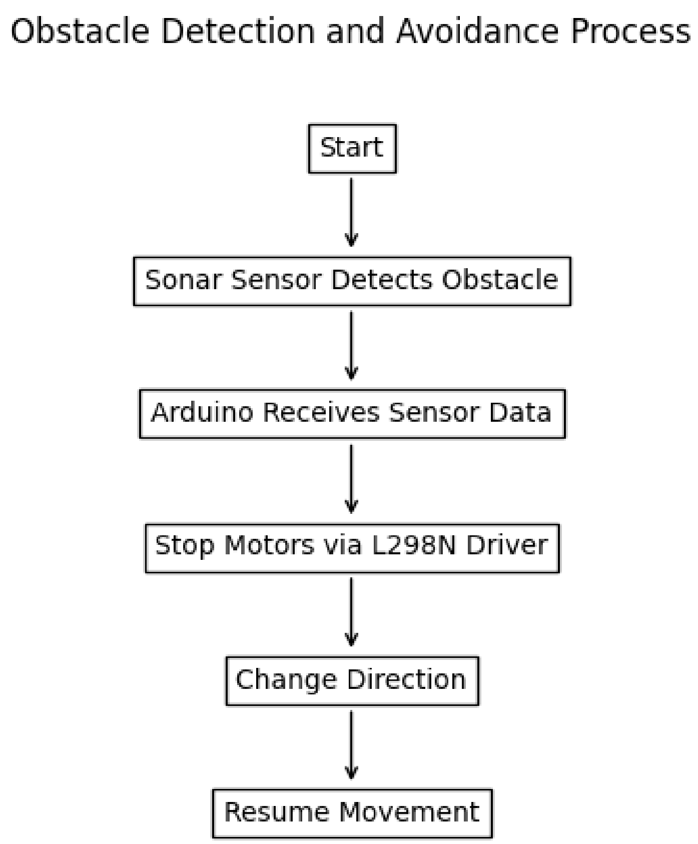


Figure 4.5: Diagram of obstacle detection and avoidance process

Facial Recognition and Intrusion Detection

- Simultaneously, the Raspberry Pi is powered on and begins receiving image data from the HD camera.
- The camera, mounted on the vehicle, captures images of any objects or persons in the vicinity.
- These images are processed by the Raspberry Pi using pre-installed libraries such as OpenCV and the face recognition library.
- The system attempts to recognize the captured faces against a pre-existing database. If a face is recognized, the system logs the recognized individual's name. If the face is not recognized, it is flagged as "Unknown," and a notification is sent indicating the presence of an unrecognized entity.

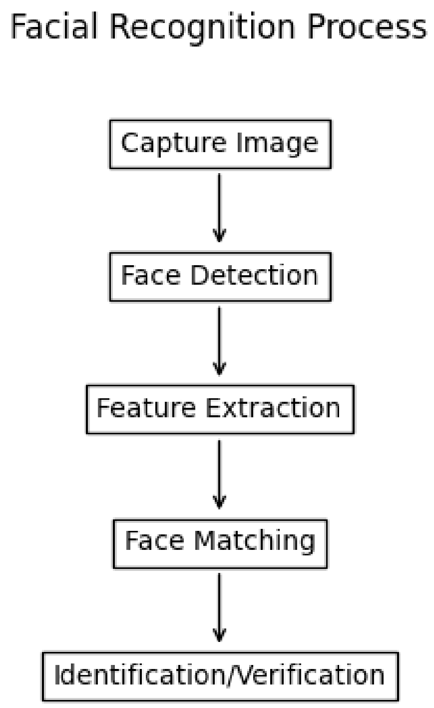


Figure 4.6: Diagram of facial recognition process

Chapter 5

Results & Analysis

5.1 Obstacle Detection & Avoidance

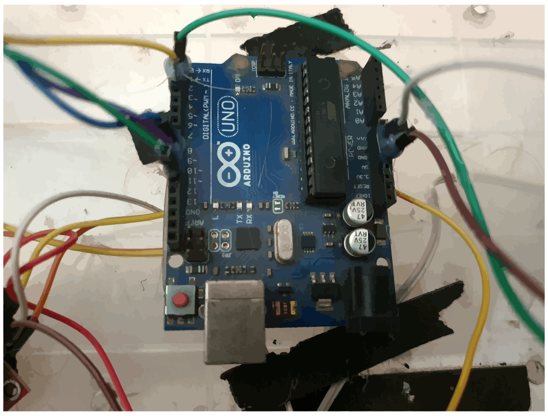
The obstacle avoidance experiment was conducted using an autonomous robot built with essential hardware components like the HC-SR04 Ultrasonic sensor, Arduino Uno, L298N motor driver, and DC motors. The robot uses a 12V Li-ion rechargeable battery to power the motors that drive the wheels, allowing it to move and rotate in various directions.

5.1.1 Working Process

- **Initialization:** The robot begins by initializing the motors and setting up the sonar sensor. The motor pins are configured using pinMode, and initial values for motor speeds are established through the setMotors function.

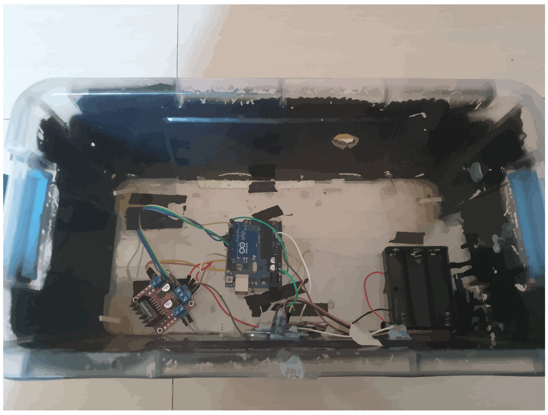
```
void setup() {
  Serial.begin(9600); // Start serial communication
  pinMode(lmf, OUTPUT);
  pinMode(lmb, OUTPUT);
  pinMode(rmf, OUTPUT);
  pinMode(rmb, OUTPUT);
  setMotors(100, 100); // Set initial motor speed
}
```

- **Movement Control and Obstacle Avoidance:** The robot is programmed to move in four stages (represented by moveDur[]), where each stage consists of forward movement followed by a 90-degree turn. During forward movement, the robot continuously checks for obstacles in front using the ultrasonic sensor. The robot moves forward for a predefined duration. If an obstacle is detected within the specified range (MAXDIST), the motors stop, and the robot pauses. After moving forward, the robot makes a 90-degree turn by running the left and right motors in opposite directions for a set time (TURN90).



(a) Arduino connection


(b) Front view


(c) Upper view

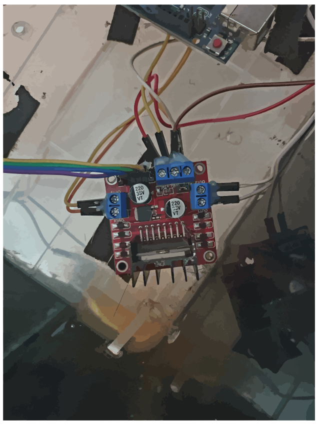

(d) L298n

Figure 5.1: Autonomous bot picture with Arduino connection

```
for (int i = 0; i < 4; i++) {
  count = 0;
  while (count < moveDur[i]) {
    setMotors(spd, spd); // Move forward

    // Stop if obstacle detected
    while (hasObject(MAXDIST)) setMotors(0, 0);
    count++;
  }
  setMotors(0, 0); // Stop
  delay(500); // Pause
  setMotors(-spd, spd); // Turn 90 degrees
  delay(TURN90); // Delay for the turn
  setMotors(0, 0); // Stop again
}

```

- **Obstacle Detection:** The ultrasonic sensor continuously checks the distance in front of the robot. If an object is detected within the MAXDIST range, the robot stops. The hasObject() function takes the maximum detection range and returns true if an obstacle is detected within that range.

```
bool hasObject(int d) {
  int distance = sonar.ping_cm();
  Serial.println(distance);
  if (distance > 3 && distance <= d) return true; // Obstacle detected
  else return false; // No obstacle
}
```

5.2 Facial Recognition

The facial recognition system was implemented on a Raspberry Pi, using the 64-bit Raspbian OS. The choice of the 64-bit OS over 32-bit was made to better leverage the Raspberry Pi's hardware capabilities, especially when handling memory-intensive tasks like image processing. The 32-bit system limits the available memory to 4 GB, while the 64-bit system can handle much more, which is beneficial for heavy image processing tasks.

Dataset Preparation A virtual environment was created for the project, and a folder named `dataset` was structured to store 15 images of the authors. These images were captured under varying conditions to test the robustness of the face recognition system. The images were organized into separate, appropriately named folders, with each folder representing a different individual.

Model Training with CNN To train the face recognition model, the `face_recognition` Python library was used. This library comes pre-trained, making it suitable for performing facial recognition without the need for training a model from scratch. Initially, the Convolutional Neural Network (CNN) model was used for training the dataset. CNNs are more accurate than the Histogram of Oriented Gradients (HOG) method, especially for image-based tasks like face recognition, due to their ability to learn complex features from data [26]. However, using CNN on the Raspberry Pi led to several issues, including long processing times, overheating, and network disturbances, making it impractical for this hardware.

As a solution, the model training and encoding were offloaded to Google Colab, a cloud-based platform, which reduced the load on the Raspberry Pi. The training model used CNN for encoding facial features, which were then saved as a file named `encodings.pickle`. The `encodings.pickle` file is a serialized object containing facial embeddings (encodings) along with corresponding names. This file is crucial for face recognition, as it stores the learned representations of faces, enabling the system to match live images against the trained dataset during the recognition stage. The encodings are computed from the facial images using the `face_recognition.face_encodings()` function and saved in the pickle file for later use.

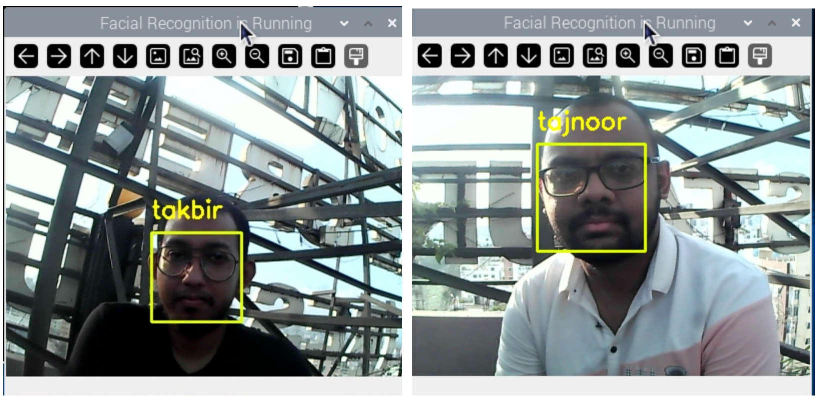
```
        image = cv2.imread(imagePath)
        rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
        # detect the (x, y)-coordinates of the bounding boxes
        # corresponding to each face in the input image
        boxes = face_recognition.face_locations(rgb,
                model="cnn")

        # compute the facial embedding for the face
        encodings = face_recognition.face_encodings(rgb, boxes)

         # loop over the encodings
        for encoding in encodings:
                # add each encoding + name to our set of known names and
                # encodings
                knownEncodings.append(encoding)
                knownNames.append(name)
```

The use of `rgb` in this section is necessary because the `face_recognition` library expects images in the RGB (Red, Green, Blue) color format for accurate face detection and encoding. By default, images loaded using OpenCV (`cv2.imread()`) are in BGR (Blue, Green, Red) format. Thus, the code converts the image from BGR to RGB using `cv2.cvtColor(image, cv2.COLOR_BGR2RGB)` before passing it to the `face_recognition` functions.

This conversion ensures that the color channels are in the correct order, allowing the face detection and encoding algorithms in the `face_recognition` library to work properly. If the image were passed in BGR format, the results could be inaccurate due to the misinterpretation of color channels.



(a) Author Takbir (b) Author Tajnoor

Figure 5.2: Face Detection and Recognition

Face Recognition with HOG While CNN was effective during the training phase, it proved inefficient for real-time face recognition on the Raspberry Pi due to the aforementioned hardware limitations. Instead, the HOG model was used for the recognition stage. HOG is computationally less intensive than CNN and can run efficiently on a Raspberry Pi, offering faster processing speeds without the risk of overheating . Although HOG may be slightly less accurate than CNN, it was a better choice for real-time recognition in this scenario.

The facial recognition system uses the `encodings.pickle` file during the recognition process. A live video stream from the HD camera is processed, and the HOG model is applied to detect faces in the frames. The system then compares the detected faces to the known encodings using `face_recognition.compare_faces()`. If a match is found, the corresponding person's name is displayed on the screen.

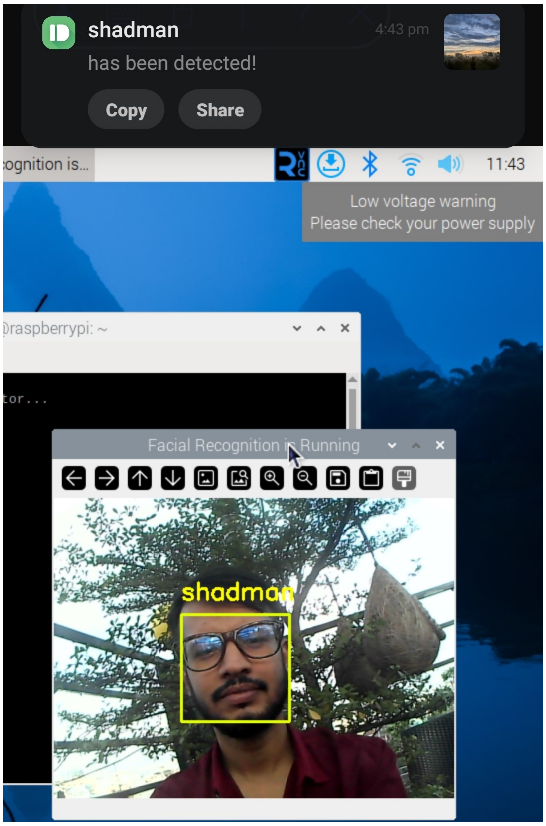


Figure 5.3: Notification for Known face Shadman

Pushbullet Notification System The system also integrates with the Pushbullet API to send notifications when a person is recognized. When a known face is detected, a single notification is sent. For unknown faces, continuous alerts are triggered to notify the user of an unrecognized entity. This functionality was implemented by maintaining a list of recognized names, ensuring notifications were not sent repeatedly for the same person.

5.3 System Testing

Obstacle Avoidance Accuracy The obstacle avoidance system was tested with various types of obstacles, each presenting different challenges in terms of detection and successful evasion. The results, summarized in Table. 5.1 demonstrate the effectiveness of the system under different conditions. The accuracy of obstacle detection and avoidance was measured by the system's ability to first detect an obstacle and then execute an appropriate maneuver to avoid it without collision.

Table 5.1: Obstacle Avoidance accuracy

Type of Obstacle	Detected	Avoided	Accuracy
Single solid thick obstacle	Yes	Yes	75%
Single solid thin obstacle	Yes	Yes	100%
Uniform shaped surface	Yes	Yes	100%
Double solid obstacles	Yes	Yes	75%
Double solid obstacles	Yes	Yes	70%

Face Recognition Accuracy The face recognition component of the system was evaluated by testing the recognition accuracy at various distances. The system was trained with a set number of images per individual and then tested for both detection and recognition success rates. The results, as presented in Table 2, provide insight into the system's performance in identifying known individuals versus signaling unknown faces.

Table 5.2: Face Recognition Accuracy

Distance (cm)	Face	Training Images	Face Detection Success Rate	Face Recognition Accuracy
30	Takbir	5	100.00%	100.00%
40	Tajnoor	5	100.00%	100.00%
50	Shadman	5	100.00%	100.00%

5.4 Discussion

The results from the obstacle avoidance and face recognition tests reveal several important insights into the system's capabilities and limitations, suggesting areas for future enhancement.

5.4.1 Obstacle Avoidance System

The obstacle avoidance tests yielded varying levels of success depending on the type and complexity of the obstacles. The system demonstrated near-perfect accuracy when dealing with single solid thin obstacles and uniform shaped surfaces, where the sonar sensor could easily detect and respond to straightforward, clear reflections. However, the performance declined with single solid thick obstacles and particularly with double solid obstacles, where the accuracy dropped to 75% and 70%, respectively. Interpretation of Results:

- *Thick Obstacles:* The slightly reduced accuracy with thick obstacles could be attributed to the way the sonar sensor interprets larger, more complex surfaces. A thick obstacle may produce more diffuse or inconsistent sonar reflections, particularly if the obstacle has an irregular shape or is not directly aligned with the sensor.

- *Double Obstacles:* The presence of multiple obstacles in close proximity introduces additional complexity. The sonar sensor, while capable of detecting the obstacles, may struggle to accurately map the environment, leading to less effective avoidance maneuvers. This suggests that the current obstacle avoidance algorithm might not be fully equipped to handle scenarios with closely spaced or overlapping objects.

Implications and Future Work The results highlight a critical area for improvement in the obstacle avoidance system. Enhancing the sensor array by incorporating additional sensors could provide more comprehensive environmental data, improving the system's ability to accurately detect and avoid complex obstacles. Moreover, refining the algorithm to better handle multiple reflections and dynamically update the obstacle map in real-time could further boost accuracy, especially in scenarios involving multiple or irregularly shaped obstacles.

5.4.2 Face Recognition System

The face recognition tests revealed a high degree of reliability, with a 100% success rate across different distances and individuals. This uniformity indicates that the system's algorithm is robust and capable of maintaining accuracy even as the distance between the camera and the subject increases. Interpretation of Results:

- *Consistency Across Distances:* The system's consistent performance at 30 cm, 40 cm, and 50 cm suggests that the face recognition algorithm is effective at processing facial features over a range of distances. This consistency is crucial for real-world applications where the subject's distance from the camera can vary.

- *Training Set Size:* The use of five training images per individual appears to be sufficient for the system to develop a reliable model of each face, leading to perfect recognition accuracy. This suggests that the algorithm is well-calibrated for the size of the training set and can generalize effectively from a small sample of images.

Implications and Future Work While the face recognition system performed flawlessly under the controlled conditions of this test, real-world environments are often less predictable. Factors such as changes in lighting, facial expressions, or the presence of multiple faces in the frame could potentially impact performance. Future work should focus on testing the system under more varied conditions to assess its robustness. Additionally, incorporating machine learning techniques that allow the system to continue learning and adapting to new faces or changing environments could further enhance its effectiveness.

The final integration combined these systems, allowing the autonomous car not only to navigate its environment by avoiding obstacles but also to recognize and alert about individuals through facial recognition. This blending of functionalities enhanced the car's ability to both interact with and respond to its surroundings dynamically.

Table 5.3: Costing Chart

Study	Obstacle Avoidance	OA Cost in USD	Face Recognition Using Pi	FR Cost in USD
This	Yes	11.44	Yes	51.45
Authors of another study [27]	Yes	21.45	No	NA
From online shop Bangladesh [28]	Yes	20.91	No	NA
From online shop India [29]	Yes	18.47	No	NA

5.4.3 Cost Effectiveness

Though there are a significant amount of works already done on each of the individual project obstacle avoidance and facial recognition, most of those studies did not provide their costing chart. Moreover, a combination of the best of both technologies was even harder to find. Furthermore, almost no study joining these two technologies provided any cost analysis, but provided the equipment lists at most cases. From those studies, it was visible that they already used high configuration devices and hence was not considered low cost already. Then this study was compared to some online ready made projects and study that could be found with cost.

From Table. 5.3, it is visible that this study already costs less than most other projects in obstacle avoidance system. For a combination of both obstacle avoidance and facial recognition, this can be considered as a significant low cost solution.

Chapter 6

Impact on Society

6.1 Sustainability of the Project/Work

Assessing Current Sustainability The integration of obstacle avoidance and intrusion detection using IoT devices has significant potential for sustainable applications, both in terms of technology and resource management. To understand its sustainability, we must assess the current state of IoT technology and its usage in the project. The lightweight sensors and low-power devices used in this project minimize energy consumption and ensure long-term functionality. Additionally, leveraging existing IoT infrastructure, such as cloud platforms, enhances the sustainability of the system by reducing hardware reliance and energy usage.

Developing a Sustainable Strategy To maintain sustainability in the long term, the project can adopt strategies for improving both hardware and software efficiencies. Upgrading to more energy-efficient sensors and optimizing algorithms for obstacle detection and facial recognition will help in reducing the system's carbon footprint. Additionally, integrating renewable energy sources such as solar panels to power the autonomous vehicle and IoT devices will further enhance the sustainability of the project.

Financial Sustainability The low-cost hardware and sensors used in this project ensure financial sustainability. The widespread adoption of IoT technology drives down costs, making the project economically viable for use in multiple sectors such as security, surveillance, and automation. Once implemented, the system's minimal operational costs will contribute to its long-term financial sustainability.

Organizational Sustainability As IoT continues to evolve, organizations can continuously integrate more advanced technology into their infrastructure. The modular nature of this system allows for scalable upgrades without the need for complete overhauls. Additionally, the data collected through IoT devices can be used for continuous improvement, enhancing system efficiency and making it adaptable to organizational needs over time.

6.2 Social and Environmental Effects and Analysis

Social Effects The integration of obstacle avoidance and intrusion detection using IoT devices has the potential to significantly impact societal security and convenience. In public spaces, autonomous vehicles equipped with obstacle detection can improve the safety of pedestrians, especially in crowded or complex environments. Additionally, the intrusion detection aspect enhances security in homes and businesses, allowing for real-time notifications and improved monitoring through facial recognition. This can help reduce crime rates and make communities safer by providing early detection and intervention.

Moreover, the implementation of this technology in smart cities can enhance urban mobility by reducing accidents and improving traffic flow. In terms of privacy, the system is designed to notify only in cases of genuine intrusion, reducing unnecessary alerts and fostering trust in surveillance systems.

Environmental Effects The project aims to utilize energy-efficient IoT devices, which minimizes their environmental impact. By incorporating sensors and cameras that operate on minimal power, the system consumes less energy than traditional surveillance or automated systems. The car's obstacle avoidance system also promotes energy efficiency by optimizing routes and reducing the need for frequent stopping or rerouting, contributing to lower emissions in automated vehicle fleets. Overall, the project aims to have minimal environmental disruption while delivering enhanced societal benefits.

6.3 Addressing Ethics and Ethical Issues

Ethical Considerations The development of IoT-based obstacle avoidance and intrusion detection systems raises several ethical concerns that must be addressed. Some key ethical aspects include:

- **Privacy Concerns:** The use of facial recognition in public or private spaces should comply with privacy regulations, ensuring that data is collected only with user consent and for specific purposes.
- **Data Security:** The system must ensure that the collected data, especially video feeds and facial recognition records, is securely stored and only accessible to authorized individuals.
- **Bias in Recognition:** Steps must be taken to avoid bias in facial recognition algorithms, ensuring fair and accurate detection for all individuals, regardless of race or gender.
- **Responsible Use of Surveillance:** The system should be used responsibly to prevent misuse of surveillance capabilities, ensuring that it serves its intended purpose of enhancing safety without compromising civil liberties.

By addressing these ethical issues, the project can ensure responsible implementation and maintain public trust in the use of IoT technologies for safety and security.

Chapter 7

Lesson Learned

7.1 Problems Faced During This Study

Throughout this study, several challenges were encountered, particularly in implementing Convolutional Neural Networks (CNNs) on the Raspberry Pi. The primary issues included excessively long processing times, overheating of the hardware, and network disturbances, which rendered the use of CNNs impractical for this low-power device. These constraints hindered the real-time performance required for effective obstacle avoidance and facial recognition, making it difficult to achieve the desired system responsiveness and reliability.

7.2 Solution to the Problems

To address these challenges, a strategic adjustment was made by offloading the computationally intensive tasks of model training and encoding to Google Colab, a cloud-based platform. This approach significantly alleviated the processing burden on the Raspberry Pi, thereby resolving issues related to overheating and slow processing times. By leveraging the cloud's robust computational resources, the system's performance was enhanced, allowing for more efficient and effective integration of the CNN-based facial recognition technology with the obstacle avoidance system.

Chapter 8

Future Work & Conclusion

8.1 Conclusion

This project successfully developed a comprehensive system that integrates obstacle avoidance and intrusion detection using IoT devices, demonstrating the potential of combining multiple technologies to enhance the functionality and security of autonomous systems. The system, which utilized ultrasonic sensors, an Arduino Uno, a Raspberry Pi 3, and facial recognition technology, showcased its ability to detect and avoid obstacles while simultaneously identifying and alerting about intrusions based on facial recognition.

The obstacle avoidance system, powered by sonar sensors, proved to be reliable in detecting objects in the vehicle's path, particularly in simple environments with single, solid obstacles. However, performance declined in more complex scenarios involving thick or closely spaced obstacles, suggesting the need for further improvements in sensor integration and algorithm refinement. This indicates that enhancing the system with additional sensors or more sophisticated algorithms could further improve accuracy in diverse environments.

The face recognition system demonstrated excellent reliability in controlled conditions, achieving a 100% success rate across different distances and individuals. This indicates that the algorithm is well-suited for environments where the subject's distance from the camera may vary, though further testing in less controlled, real-world conditions is recommended. Integrating machine learning for continuous system adaptation could improve its robustness against challenges like changing lighting conditions or facial expressions.

The final integration of both systems allowed the autonomous car to not only navigate its environment but also to monitor for unauthorized individuals using facial recognition. This dual functionality significantly improves the vehicle's real-time decision-making capabilities, making it well-suited for security and surveillance applications.

In terms of cost-effectiveness, the project demonstrated that a functional, reliable system could be developed using readily available, low-cost components, making it accessible for various practical applications. Compared to other studies, this solution provided a more affordable approach to combining obstacle avoidance and facial recognition.

In conclusion, this project lays the foundation for further advancements in the fields of autonomous navigation and IoT-based security systems. By addressing the limitations identified in testing, future iterations can improve the system's accuracy, adaptability, and performance in real-world conditions. This work offers valuable insights and contributes to the ongoing development of intelligent, secure autonomous systems for various industries, including robotics, smart vehicles, and home automation.

8.2 Future Works

In future work, the system could benefit from enhanced sensor integration, such as adding infrared or lidar sensors, to improve accuracy in detecting complex or irregularly shaped obstacles. Additionally, optimizing the obstacle detection algorithms to handle multi-obstacle scenarios more effectively would increase the system's adaptability in dynamic environments. The facial recognition component could be further tested under diverse real-world conditions, including changes in lighting, facial expressions, and the presence of multiple faces, to ensure robustness. Implementing adaptive machine learning techniques could allow the system to continuously improve its recognition capabilities. Expanding the system's IoT functionality, such as adding cloud-based remote monitoring and control, would make it more versatile for security and surveillance applications. Lastly, exploring energy-efficient power solutions to extend the battery life could enhance the system's practicality for long-term autonomous operation.

Bibliography

[1] M. S. M. Rathod and M. S. Apte, “Obstacle detection and avoidance for autonomous electric vehicle using arduino,” *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 1, 2019.

[2] M. Mira, S. Hadi, A. R. Fachrudin, S. H. Susilo, and R. E. Perkasa, “Vehicle safety system with arduino-based face detection technique,” *Attractive: Innovative Education Journal*, vol. 3, no. 3, pp. 196–202, 2021.

[3] J. Azeta, C. Bolu, D. Hinvi, and A. A. Abioye, “Obstacle detection using ultrasonic sensor for a mobile robot,” in *IOP Conference Series: Materials Science and Engineering*, vol. 707, p. 012012, IOP Publishing, 2019.

[4] A. Varma, A. Ashwath, A. Verma, A. Bagubali, and K. V. Krishnan, “Implementation of obstruction avoiding robot using ultrasonic sensor and arduino uno.”

[5] A. Goswami, B. Bhattacharjee, R. Debnath, and A. Sikder, “Analysis of obstacle detection with distance measuring using arduino uno and ultrasonic sensor,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 11, 2021.

[6] S. Adarsh, S. M. Kaleemuddin, D. Bose, and K. Ramachandran, “Performance comparison of infrared and ultrasonic sensors for obstacles of different materials in vehicle/robot navigation applications,” in *IOP Conference Series: Materials Science and Engineering*, vol. 149, p. 012141, IOP publishing, 2016.

[7] B. Pavithra, P. S. S. Rao, A. Sharmila, S. Raja, and S. Sushma, “Characteristics of different sensors used for distance measurement,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 12, pp. 698–702, 2017.

[8] J. Z. Varghese, R. G. Boone, *et al.*, “Overview of autonomous vehicle sensors and systems,” in *International Conference on Operations Excellence and Service Engineering*, vol. 2015, sn, 2015.

[9] R. Dominguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzalez, “Lidar based perception solution for autonomous vehicles,” in *2011 11th International Conference on Intelligent Systems Design and Applications*, pp. 790–795, IEEE, 2011.

[10] J. Zhou, H. Chen, and C. Xiu, “A simulation model to evaluate and verify functions of autonomous vehicle based on simulink®,” in *Intelligent Robotics and Applications: Second*

International Conference, ICIRA 2009, Singapore, December 16-18, 2009. Proceedings 2, pp. 645–656, Springer, 2009.

[11] H. K. Heidarsson and G. S. Sukhatme, "Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar," in *2011 IEEE International Conference on Robotics and Automation*, pp. 731–736, IEEE, 2011.

[12] M. Gangawane, R. Awate, R. Suryawanshi, and R. Joshi, "Obstacle detection and object size measurement for autonomous mobile robot using sensor," *Int. Res. J. Eng. Technol.(IRJET)*, vol. 3, no. 05, 2016.

[13] M. Sharif, F. Naz, M. Yasmin, M. A. Shahid, and A. Rehman, "Face recognition: A survey.," *Journal of Engineering Science & Technology Review*, vol. 10, no. 2, 2017.

[14] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, "Face recognition systems: A survey," *Sensors*, vol. 20, no. 2, p. 342, 2020.

[15] R. Rosnelly, M. S. Simanjuntak, A. C. Sitepu, M. Azhari, S. Kosasi, *et al.*, "Face recognition using eigenface algorithm on laptop camera," in *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1–4, IEEE, 2020.

[16] T. Mantoro, M. A. Ayu, *et al.*, "Multi-faces recognition process using haar cascades and eigenface methods," in *2018 6th international conference on multimedia computing and systems (ICMCS)*, pp. 1–5, IEEE, 2018.

[17] T. H. Le, "Applying artificial neural networks for face recognition," *Advances in Artificial Neural Systems*, vol. 2011, no. 1, p. 673016, 2011.

[18] M. Pilgrim and S. Willison, *Dive Into Python 3*, vol. 2. Springer, 2009.

[19] G. Bradski, A. Kaehler, *et al.*, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, no. 2, 2000.

[20] K. D. Ismael and S. Irina, "Face recognition using viola-jones depending on python," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 3, pp. 1513–1521, 2020.

[21] P. G. MAR-HERNÁNDEZ, P. L. IBARRA-ANGULO, J. C. GRIJALVA-ACUÑA, and J. H. ABRIL-GARCÍA, "Image recognition software geometry with python and opencv.," *Journal Computer Technology*, vol. 7, no. 19, 2023.

[22] M. Pilgrim, "Serializing python objects," in *Dive Into Python 3*, pp. 205–223, Springer, 2009.

[23] B. Arduino, "Arduino uno," *Datasheet.(Downloaded: 13 Jun 2020). https://datasheet. octopart. com/A000066-Arduino-datasheet-38879526. pdf*, 2015.

[24] E. Upton and G. Halfacree, *Raspberry Pi user guide*. John Wiley & Sons, 2016.

[25] S. Vijayalakshmi *et al.*, "Robotic car using arduino with bluetooth controller," *Ijise*, 2019.

[26] M. F. Aslan, A. Durdu, K. Sabanci, and M. A. Mutluer, "Cnn and hog based comparison study for complete occlusion handling in human tracking," *Measurement*, vol. 158, p. 107704, 2020.

[27] S. S. Goswami and S. K. Sahoo, "Design of a robotic vehicle to avoid obstacle using arduino microcontroller and ultrasonic sensor," in *Applied Mathematics, Modeling and Computer Simulation*, pp. 1002–1010, IOS Press, 2023.

[28] RoboticsBD.com, "Arduino Obstacle Avoiding Robot Kit Bangladesh — store.roboticsbd.com." `https://store.roboticsbd.com/robotics-parts/127-obstacle-avoiding-robot-kit-bangladesh.html`. [Accessed 12-09-2024].

[29] indiamart.com, "indiamart.com." `https://www.indiamart.com/proddetail/obstacle-avoiding-robot-kit-with-arduino-21733204291.html`. [Accessed 12-09-2024].

Cost Effective IoT-Based Smart System for Avoidance of Obstacle and Intruder Detection

By

Md. Shadman Sakib Talukdar, Mahiyan Rahman Takbir, Imam Tajnoor Hossain Amrit

Supervised By

Mohammad Noor Nabi
Senior Lecturer
Department of Computer Science & Engineering
School of Engineering, Technology & Sciences
Independent University, Bangladesh

Summer, 2024

Accepted in International Conference on Intelligent Cybernetics Technology and Applications 2024
Will be published In IEEE Xplore

Cost Effective IoT-Based Smart System for Avoidance of Obstacle and Intruder Detection

By

Md. Shadman Sakib Talukdar, Mahiyan Rahman Takbir, Imam Tajnoor Hossain Amrit

Summer, 2024

The student modified the senior project final report as per the recommendation made by their academic supervisor and/or panel members during final viva, and the department can use this version for archiving.

Signature of the Supervisor

Mohammad Noor Nabi
Senior Lecturer
Department of Computer Science & Engineering
School of Engineering, Technology & Sciences
Independent University, Bangladesh